



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ODHAD RYCHLOSTI VOZIDLA ZE ZÁZNAMU ON-BOARD KAMERY

VEHICLE SPEED ESTIMATION FROM ON-BOARD CAMERA RECORDING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. KRYŠTOF JANÍČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAKUB ŠPAŇHEL

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2017/2018

Zadání diplomové práce

Řešitel: **Janíček Kryštof, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Odhad rychlosti vozidla ze záznamu on-board kamery**
Vehicle Speed Estimation from On-Board Camera Recording

Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte dostupné materiály ohledně problematiky odhadu rychlosti vozidla ze záznamu s využitím vizuální odometrie.
2. Zorientujte se v současných metodách odhadu rychlosti vozidla ze záznamu palubní kamery automobilu.
3. Vyberte vhodnou metodu a navrhnete možné řešení dané problematiky.
4. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
5. Vaše řešení řádně vyhodnoťte a porovnejte dosažené výsledky různých metod; diskutujte možnosti budoucího vývoje.
6. Vytvořte stručný plakát a video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

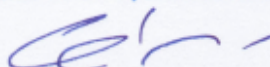
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Špaňhel Jakub, Ing.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 23. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá návrhem a vývojem systému pro odhad rychlosti vozidla ze záznamu palubní kamery. Odhad rychlosti je založen na odhadu optického toku mezi dvěma snímky a konvoluční neuronové síti. Na vytvořené datové sadě dokáže navržený systém odhadnout rychlost s průměrnou chybou 20% při rychlostech vyšších než 35 kilometrů za hodinu.

Abstract

This thesis describes the design and implementation of system for vehicle speed estimation from on-board camera recording. Speed estimation is based on optical flow estimation and convolutional neural network. Designed system is able to estimate speed with average error of 20% on created data set where actual speed is greater than 35 kilometers per hour.

Klíčová slova

odhad rychlosti, palubní kamera, video záznam, vizuální odometrie, konvoluční neuronová síť, optický tok

Keywords

speed estimation, on-board camera, camera recording, visual odometry, convolutional neural network, optical flow

Citace

JANÍČEK, Kryštof. *Odhad rychlosti vozidla ze záznamu on-board kamery*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jakub Špaňhel

Odhad rychlosti vozidla ze záznamu on-board kamery

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jakuba Špaňhela. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Kryštof Janíček
21. května 2018

Poděkování

Rád bych poděkoval panu Ing. Jakubovi Špaňhelovi za odborné vedení práce, cenné rady a vstřícnost při konzultacích. Dále bych rád poděkoval své rodině za podporu po dobu studia.

Obsah

| | | |
|----------|---|-----------|
| 1 | Úvod | 2 |
| 2 | Odhad rychlosti vozidla ze záznamu palubní kamery | 3 |
| 2.1 | Rychlost | 3 |
| 2.2 | Palubní kamera | 3 |
| 2.3 | Vizuální odometrie | 4 |
| 2.4 | Existující řešení odhadu rychlosti z palubní kamery | 4 |
| 2.5 | Využití systému pro odhad rychlosti vozidla | 6 |
| 3 | Metody použité v této práci | 7 |
| 3.1 | Optický tok | 7 |
| 3.2 | Konvoluční neuronové sítě | 9 |
| 3.3 | Rekurentní neuronové sítě | 12 |
| 4 | Návrh systému | 16 |
| 4.1 | Dekompozice systému | 16 |
| 4.2 | Omezení systému | 18 |
| 5 | Implementace systému | 19 |
| 5.1 | Vytvořená datová sada | 19 |
| 5.2 | Optický tok | 22 |
| 5.3 | Trénování konvoluční neuronové sítě | 22 |
| 5.4 | Odhad rychlosti | 23 |
| 5.5 | Validace natrénované neuronové sítě | 24 |
| 5.6 | Použité technologie | 25 |
| 6 | Experimenty a vyhodnocení systému | 27 |
| 6.1 | Vstup konvoluční neuronové sítě | 27 |
| 6.2 | Rozšíření sítě o jednotku LSTM | 28 |
| 6.3 | Filtrace šumu v optickém toku | 29 |
| 6.4 | Vyhodnocení systému | 33 |
| 7 | Závěr | 40 |
| | Literatura | 41 |

Kapitola 1

Úvod

V současné době velkého rozmachu nejrůznějších systémů z oblasti informačních technologií se tyto systémy stávají i součástí automobilů. Téměř v každém moderním vozidle lze dnes najít i videokameru, která snímá události v okolí vozidla. Na základě těchto informací je pak řidiči poskytováno množství dat či dokonce různé bezpečnostní prvky. Jednou z těchto informací může být i odhad rychlosti vozidla.

Cílem této práce bylo prostudovat různé metody používané pro odhad rychlosti vozidla ze záznamu palubní kamery, navrhnout řešení daného problému a následná implementace tohoto systému. A nakonec navržený systém důsledně vyhodnotit a diskutovat další možné kroky jeho vývoje.

Tato práce je strukturována následovně. Úvod práce následuje kapitola 2, ve které jsou čtenáři přiblíženy pojmy jako rychlost, palubní kamera či vizuální odometrie. Následuje stručný průzkum používaných postupů pro odhad rychlosti vozidla ze záznamu palubní kamery. Na závěr této kapitoly je diskutováno možné využití daného systému.

Kapitola 3 se zabývá detailnějším popisem vybraných metod, které jsou použity v této práci. V kapitole 4 je rozebrán návrh systému a jeho omezení, která vyplývají z použitých metod.

Implementace systému je rozebrána v kapitole 5. Je zde popsána vytvořená datová sada, její pořízení a zpracování. V dalších částech je rozebrána samotná implementace systému a její validace. Závěrem kapitoly jsou stručně popsány použité technologie.

Předposlední částí práce je kapitola 6, ve které jsou popsány experimenty provedené za účelem vylepšení odhadu rychlosti. Druhá část této kapitoly se věnuje detailnímu vyhodnocení systému, diskuzi problematických situací a možnostem dalšího vývoje této práce.

Závěrem práce je kapitola 7, kde jsou stručně shrnuty výsledky této práce.

Kapitola 2

Odhad rychlosti vozidla ze záznamu palubní kamery

V této kapitole jsou stručně popsány základní poznatky o rychlosti a dále seznámení se s pojmy palubní kamera a vizuální odometrie. Další sekce stručně zkoumá současné existující metody odhadu rychlosti vozidla ze záznamu palubní kamery. V poslední sekci je rozebráno případné využití takového systému.

2.1 Rychlost

Rychlost je charakteristika pohybu, která nám sděluje, jakým způsobem se mění poloha tělesa v čase. Velikost okamžité rychlosti v hmotného bodu v čase t definujeme jako podíl přírůstku dráhy Δs , k němuž dojde za čas Δt , a této doby:

$$v = \frac{\Delta s}{\Delta t}, [v] = \text{m.s}^{-1} \quad (2.1)$$

Okamžitá rychlost je fyzikální veličina, která má vždy směr tečny k dané trajektorii hmotného bodu a je orientována ve směru změny uražené dráhy. Při pohybu hmotného bodu po křivce se vždy mění směr okamžité rychlosti, ale nemusí se měnit velikost rychlosti [20].

2.2 Palubní kamera

Palubní kamera je zařízení, které je ve většině případů umístěné na čelním skle vozidla a zaznamenává události před vozidlem. Záznam je zpravidla ukládán na paměťovou kartu pro další použití. Tato zařízení jsou v poslední době stále populárnější a jejich záznam může sloužit jako důkazní prostředek v případě dopravní nehody či jiného incidentu. Například v Ruské federaci je tento záznam u soudů upřednostňován před očitým svědectvím, ale například v Rakousku jsou tato zařízení v soukromých vozidlech zcela zakázána [24].

V současnosti je možné zakoupit palubní kameru velice levně, avšak existují i dražší modely. Cena je dána především množstvím funkcí daného modelu. Základní vlastnosti vycházejí z běžných kamer a jsou jimi například rozlišení obrazu či velikost paměti. Mezi pokročilejší funkce palubních kamer patří GPS senzor, který zaznamenává polohu, G-senzor, který snímá zrychlení vozidla, automatické zapnutí a vypnutí při detekci pohybu a další.

Pro tyto účely je možné využít i „chytrý telefon“ a to například prostřednictvím aplikace AutoGuard Blackbox¹ či CaroO Free².

2.3 Vizualní odometrie

Vizualní odometrie je proces odhadu pózy daného prostředku (např. robot, vozidlo či člověk), který je založen pouze na sekvenci obrazových dat zachycených pomocí jedné či více kamer, které jsou připevněny k danému prostředku. Jádrem je tedy odhad pózy samotné kamery. Vizualní odometrii lze také chápat jako odhad vlastního pohybu (*ego-motion*) kamery v 3D prostoru [1].

Vizualní odometrie je typicky používána pro navigaci robotů v neznámém prostředí. Mezi nejznámější oblasti užití patří pravděpodobně navigace vesmírných robotů jako je Mars rover.

Aktuální přístup k vizualní odometrii je představen v článku [22]. Autoři zde odhadují vlastní pohyb kamery pomocí rekurentní konvoluční neuronové sítě. Autoři uvádějí, že dosahují obdobných výsledků jako klasické metody. Výhodou však je, že není potřeba žádné kalibrace kamery či dalších modulů.

Metody vizualní odometrie lze rozdělit do tří kategorií – založené na význačných rysech, založené na vzhledu a na hybridní.

Metody založené na významných rysech spočívají v extrakci významných příznaků (např. rohy, čáry či křivky) v sekvenci snímků, jejich následném sledování a nakonec odhadu pohybu.

Metody založené na vzhledu se zaměřují na změny v intenzitě pixelů. Pohyb kamery může být odhadnut například použitím optického toku. Nejčastěji používanou metodou je porovnání se vzorem. V aktuálním snímku je vybrána oblast zájmu, která slouží jako vzor. Tento vzor je poté porovnáván s oblastmi v následujícím snímku a je hledána nejlepší shoda. Po nalezení odpovídající oblasti je již jednoduché určit pohyb mezi danými snímky.

Hybridní metody jsou pak založené na kombinaci metod uvedených výše. Jsou vhodné především při scénách s výraznou texturou.

2.4 Existující řešení odhadu rychlosti z palubní kamery

V této části jsou uvedeny metody používané pro odhad rychlosti vozidla ze záznamu palubní kamery. Tyto postupy lze rozdělit do dvou kategorií. První z nich se snaží získat rozdíl dráhy mezi snímky a následné dosazení do vzorce (2.1). Druhá kategorie pak obsahuje další různé přístupy.

2.4.1 Vyhledávání známých objektů

Jedním z nejběžnějších způsobů odhadu rychlosti vozidla je vyhledávání známých objektů jako jsou pouliční světla, vodorovné dopravní značení či dopravní značky. Vyhledávání těchto objektů probíhá nad každým snímkem záznamu a po nalezení objektu v sousedních

¹<https://play.google.com/store/apps/details?id=com.hovans.autoguard>

²<https://play.google.com/store/apps/details?id=com.pokevian.skids>

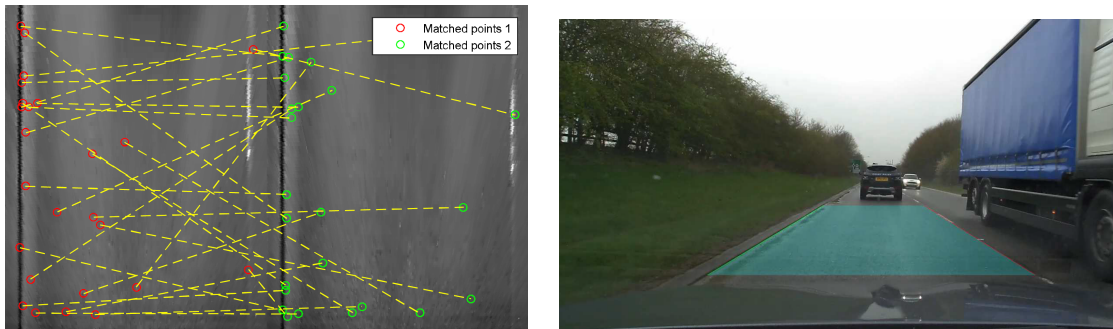
snímcích se vypočítá uražená vzdálenost. Poté se na základě časových značek jednotlivých snímků vypočítá rychlost vozidla. Tato metoda je určena především pro rovné úseky cesty. V článku [10] autor prezentuje modifikaci této metody, která je vhodná i při změně směru vozidla.

Použití této metody je však značně omezeno díky použití známých objektů, které v mnoha případech na silnicích chybí a dokonce i v případech, kdy se hledaný objekt v obraze nachází, nemusí splňovat předpokládané vlastnosti předepsané zákonem.

2.4.2 Sledování povrchu vozovky

Autoři článku [3] vycházejí z metody, která se používá pro odhad rychlosti malých bezpilotních letounů (*UAV, unmanned aerial vehicle*). Tato metoda spočívá ve sledování povrchu země, vyhledávání významných bodů a sledování jejich posunu. Pro využití v silničním vozidle je v záznamu palubní kamery vybrána oblast zájmu, která zachycuje úsek silnice těsně před vozidlem, jak lze vidět na obrázku 2.1. Tato oblast se poté transformuje do dvou-rozměrné roviny, kde jsou vyhledávány významné body. Tyto významné body jsou poté spárovány pomocí algoritmu KLT, BRISK nebo SURF v následujícím snímku záznamu, což je znázorněno na obrázku 2.1. Následně je vypočítána vzdálenost mezi odpovídajícími body na obou snímcích, která je převedena na uraženou vzdálenost vozidla. Výpočet rychlosti vozidla je pak proveden pomocí rovnice (2.1).

Tato metoda má několik omezení, kterými jsou potřeba rovného terénu před vozidlem, specifický úhel a umístění kamery a potřeba dobře rozlišitelných bodů na vozovce. Další omezení může vyplývat z použité kamery a rychlosti vozidla. Při vyšších rychlostech nemusí být možné získat použitelný snímek vozovky.



Obrázek 2.1: Sledování významných bodů vlevo a výběr oblasti zájmu vpravo pro metodu sledování povrchu vozovky. Převzato z [3].

2.4.3 Výpočet rychlosti z optického toku

V článku [2] autoři odhadují rychlost vozidla pomocí optického toku, který pomocí vektorů udává patrný pohyb zájmových bodů v obraze. Při předpokladu, že vozidlo se pohybuje po rovném povrchu a směr pohybu vozidla je paralelní s optickou osou kamery, platí:

$$u = \frac{\omega}{f}x^2 + \frac{V}{hf}xy + \omega f \quad (2.2)$$

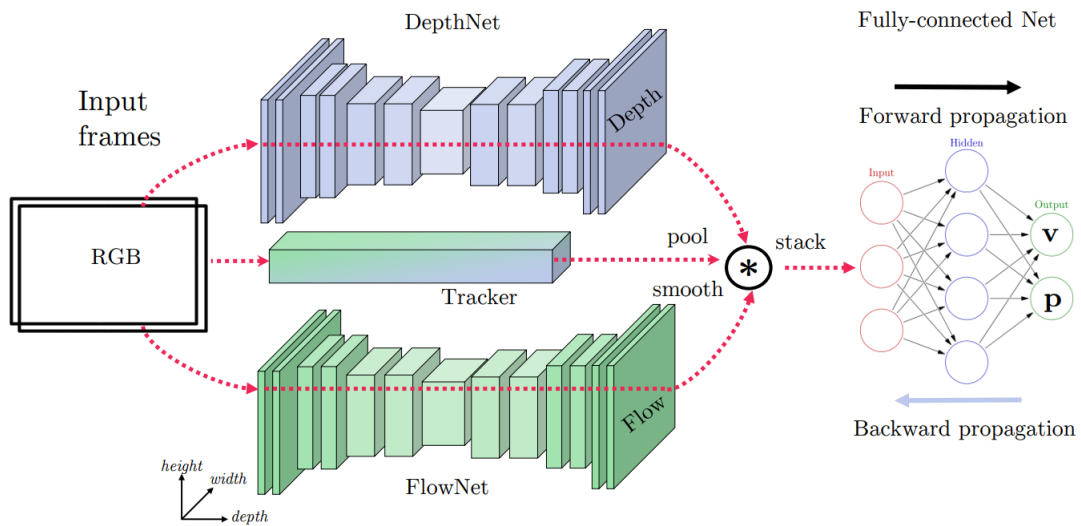
$$v = \frac{\omega}{f}xy + \frac{V}{hf}y^2 \quad (2.3)$$

kde f je ohnisková vzdálenost kamery, h je vzdálenost vozovky od umístění kamery, ω je úhlová rychlost, V je rychlost, x a y jsou souřadnice bodu v obraze.

Studie [6] se zabývá obdobným principem, avšak kamera je umístěna tak, že snímá povrch vozovky. Důležité je také poznamenat, že experimenty jsou prováděny na podkladu z písku či šterku, kde lze předpokládat výraznější strukturu oproti klasickým povrchům vozovek.

2.4.4 Odhad rychlosti pomocí konvoluční neuronové sítě

Autoři prací [14, 17] pro odhad rychlosti využívají optický tok a konvoluční neuronové sítě. V prvním zmíněném je odhadována rychlost a vzdálenost vozidel viděných na záznamu. Řešením je kombinace sítí pro odhad optického toku, hloubky v obraze a vyhledávání vozidel, celková architektura je zobrazena na obrázku 2.2. Samotný odhad rychlosti je tedy založen na konvoluční neuronové síti pro odhad optického toku a následný odhad rychlosti.



Obrázek 2.2: Ukázka architektury konvoluční neuronové sítě pro odhad rychlosti vozidla. Převzato z [14].

2.5 Využití systému pro odhad rychlosti vozidla

Lze předpokládat, že v brzké budoucnosti bude každé silniční vozidlo vybaveno kamerou či kamerami, které budou zaznamenávat okolní dění. Pořízený záznam bude důkladně prozkoumán metodami z oblasti počítačového vidění a získané informace poté zobrazeny řidiči vozidla či automaticky zpracovány. Jedním z těchto systémů může být i odhad rychlosti vozidla, což povede k nadbytečnosti fyzických senzorů pro měření rychlosti automobilu a tím pádem zlevnění jejich výroby.

V současnosti lze tento systém využít například ke kontrole profesionálních řidičů (např. nejryznější kurýři, přepravci). Zaměstnavatel pak může zjistit, zda daný řidič porušuje zákony, aby dodržel termín dodání či nikoliv.

Kapitola 3

Metody použité v této práci

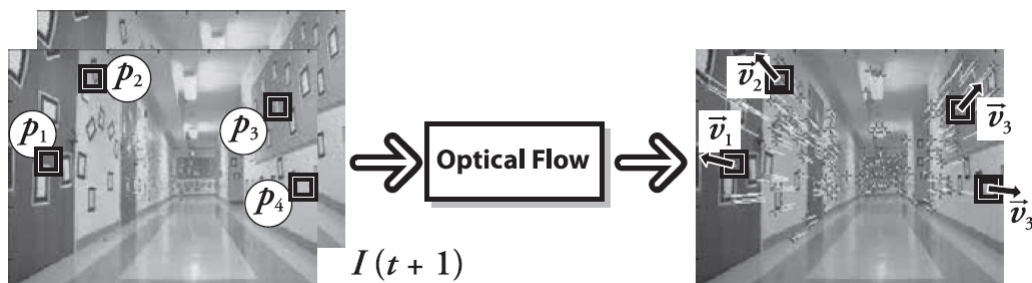
Tato kapitola se zabývá detailnějším popisem vybraných metod použitých při řešení systému pro odhad rychlosti vozidla ze záznamu palubní kamery. V první sekci je rozebrán optický tok a algoritmus pro jeho výpočet. V další sekci jsou pak detailněji popsány konvoluční neuronové sítě, jejich architektura a princip učení. Závěrečná sekce se věnuje popisu rekurentních neuronových sítí a jednotce LSTM.

3.1 Optický tok

Optický tok je metoda, která dokáže odhadnout patrný pohyb mezi dvěma následujícími snímky z obrazové sekvence. Metoda optického toku tedy vytvoří dvourozměrné pole vektorů, které udávají posun pixelů [5, 19, 23].

Určování optického toku pro každý pixel snímku se nazývá hustý optický tok. Alternativou je řídký optický tok, který určuje hodnotu posunu pouze pro malou skupinku vybraných pixelů. Tuto skupinu pixelů je potřeba předem definovat například pomocí některého z hranových detektorů. Nejznámějším algoritmem pro výpočet řídkého optického toku je Lucas-Kanade.

Běžným předpokladem v určování optického toku je konstantní hodnota jasu. Ta říká, že hodnota odpovídajících pixelů ve dvou po sobě následujících snímcích je neměnná. Ovšem ne všechny pohyby způsobí změnu jasu pixelů a ne všechny změny jasu jsou pohyby. Příkladem může být pohyb čistého papíru po stole, kdy ke změně hodnot jasu dojde pouze na okrajích listu, zatímco uvnitř je hodnota jasu konstantní. To pak vede ke zhoršení odhadu optického toku.



Obrázek 3.1: Ilustrace odhadu optického toku. Vlevo dva vstupní snímky s vyhledanými body, vpravo odhad pohybu daných bodů. Převzato z [5].

Optický tok pomocí hustého zpětného prohledávání

V této práci je použit optický tok využívající husté zpětné prohledávání (*DIS, Optical Flow using Dense Inverse Search*) [16].

Základní komponentou tohoto algoritmu je rychlé a efektivní nalezení korespondujících oblastí na dvou snímcích. Pro vzorovou oblast T ve výchozím snímku I_t o velikosti $\theta_{ps} \times \theta_{ps}$ pixelů a středem $\mathbf{x} = (x, y)^T$ je v prohledávaném snímku I_{t+1} nalezena nejlépe odpovídající oblast o velikosti $\theta_{ps} \times \theta_{ps}$ pomocí algoritmu klesání dle gradientu (*gradient descent*). Cílem je nalezení vektoru posunu $\mathbf{u} = (u, v)$ takového, že součet mocnin rozdílů mezi vzorovou a prohledávanou oblastí bude co nejmenší:

$$\mathbf{u} = \underset{\mathbf{u}'}{\operatorname{argmin}} \sum_x [I_{t+1}(\mathbf{x} + \mathbf{u}') - T(\mathbf{x})]^2 \quad (3.1)$$

Minimalizování této hodnoty není lineární a je optimalizováno iterativně použitím inverzního Lucas-Kanade algoritmu. V této metodě je použita alternace dvou kroků po určitý počet iterací nebo dokud hodnota 3.1 nekonverguje. V prvním kroku je hodnota 3.2 minimalizována pomocí současného odhadu \mathbf{u} na novou hodnotu vektoru $\Delta \mathbf{u}$ tak, že:

$$\Delta \mathbf{u} = \underset{\Delta \mathbf{u}'}{\operatorname{argmin}} \sum_x [I_{t+1}(\mathbf{x} + \mathbf{u} + \Delta \mathbf{u}') - T(\mathbf{x})]^2 \quad (3.2)$$

První krok musí získat pozici okna $I_{t+1}(\mathbf{x} + \mathbf{u})$ za použití bilineární interpolace pro dosažení přesných hodnot posunu. Druhý krok pak upřesňuje hodnotu posunu $\mathbf{u} \leftarrow \mathbf{u} + \Delta \mathbf{u}$.

Algoritmus je prováděn nad pyramidou různých rozlišení. Začíná se na nejhrubějším rozlišení θ_{ss} a postupně se prochází pyramidou až k nejjemnějšímu rozlišení θ_{sf} . Na každém stupni je provedeno pět kroků, které jsou uvedeny v algoritmu 1. Výstupem každé iterace s je pole hustého optického toku \mathbf{U}_s .

Algorithm 1 Dense Inverse Search (DIS) [16]

- 1: Set initial flow field $\mathbf{U}_{\theta_{ss}+1} \leftarrow 0$
 - 2: **for** $s = \theta_{ss}$ to θ_{sf} **do**
 - 3: Create uniform grid of N_s patches
 - 4: Initialize displacements from \mathbf{U}_{s+1}
 - 5: **for** $i = 1$ to N_s **do**
 - 6: Inverse search for patch i
 - 7: **end for**
 - 8: Densification: Compute dense flow field \mathbf{U}_s
 - 9: Variational refinement of \mathbf{U}_s
 - 10: **end for**
-



Obrázek 3.2: Odhad optického toku metodou Dense Inverse Search nad uměle vytvořenými daty. Vlevo ground truth, vpravo odhad. Převzato z [16].

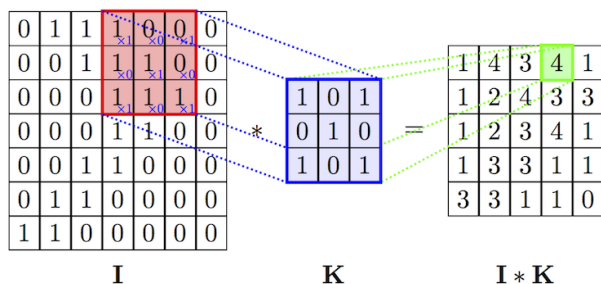
3.2 Konvoluční neuronové sítě

Konvoluční neuronové sítě jsou tvořeny neurony, které se dokáží učit pomocí vah. Každý neuron má vstup, na kterém provádí skalární součin, což je jeho výstupem. Celá síť pak dokáže převést vstupní hodnoty, většinou hodnoty pixelů obrázku, na jednu výstupní hodnotu, která reprezentuje příslušnost k nějaké třídě. Konvoluční neuronové sítě předpokládají, že vstupem je obrázek, což umožňuje zakódovat do architektury sítě určité vlastnosti. Tyto vlastnosti umožňují efektivnější dopředný průchod sítí a značně redukuje počet parametrů v síti [15].

Konvoluční neuronové sítě jsou modelovány pomocí sekvencí vrstev. V současnosti se používají tři hlavní typy vrstev při modelování architektury: konvoluční vrstva (*convolutional layer*), podvzorkovací vrstva (*pooling layer*) a plně propojená vrstva (*fully-connected layer*).

3.2.1 Konvoluční vrstva

Konvoluční vrstva je stěžejním stavebním blokem konvolučních neuronových sítí, jelikož vykonává nejvíce výpočtů. Konvoluční vrstva sestává z řady naučitelných filtrů, které jsou na výšku i šířku rozměrově malé, ale prochází celou škálou hloubky. Během dopředného průchodu je každý filtr posouván podél šířky a výšky vstupu a je vypočítán skalární součin z hodnot filtru a vstupu na dané pozici. Tento postup vytvoří dvourozměrnou aktivační mapu, která udává odezvy filtrů v každé pozici. Tyto aktivační mapy jsou poté skládány za sebou podél osy hloubky [15].



Obrázek 3.3: Grafické znázornění principu konvoluce vstupní matice I s jádrem K . Převzato z [21].

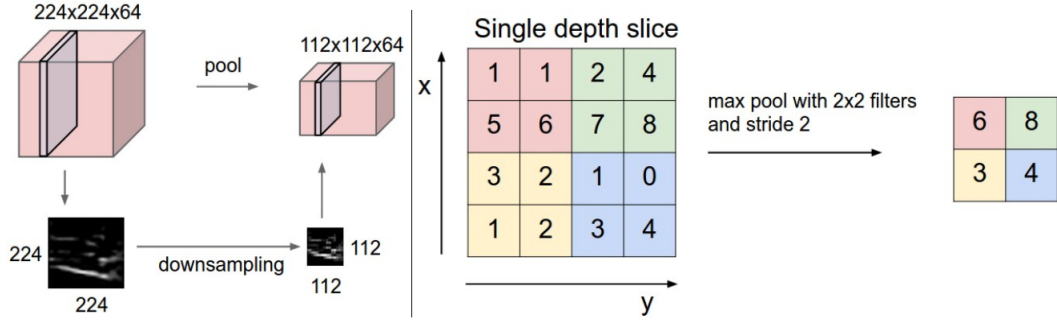
3.2.2 Podvzorkovací vrstva

Vrstva podvzorkování má za úkol zmenšit dimenze dané reprezentace obrázku, což vede ke snížení parametrů sítě a urychlení výpočtu. Podvzorkování probíhá nezávisle na každé úrovni hloubky. Pro výběr vzorků je nejčastěji používána funkce maxima, ale existují i další funkce jako je průměr nebo L2-norm.

Pro podvzorkování se nejčastěji používá filtr o velikosti 2×2 s krokem 2, což zahodí 75% aktivací. Tedy při použití těchto velikostí a funkce pro výběr maxima jsou vybírány maximální hodnoty v oblasti 2×2 pixely a ty jsou zachovány na výstupu. Jednotlivé procházené oblasti jsou od sebe pak odsazeny o velikost kroku. Tento princip je znázorněn na obrázku 3.4. Výstupní dimenzi lze spočítat dle vzorců (3.3), kde W_1 je šířka vstupu,

H_1 je výška vstupu, D_1 je hloubka vstupu, W_2 je šířka výstupu, H_2 je výška výstupu, D_2 je hloubka výstupu, F je velikost filtru a S je velikost kroku [15].

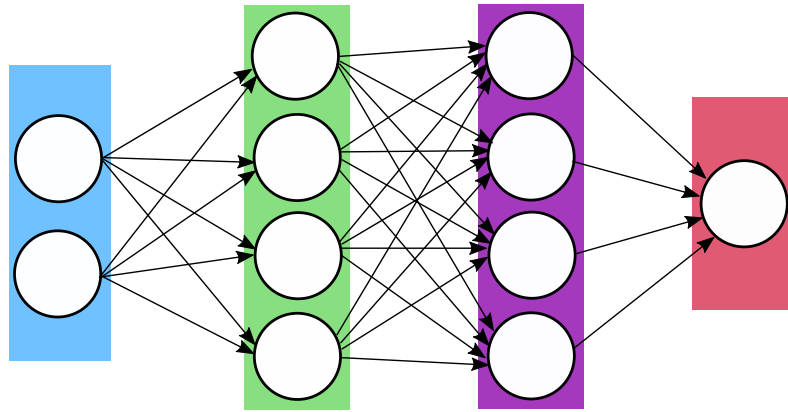
$$W_2 = \frac{(W_1 - F)}{S + 1}, H_2 = \frac{(H_1 - F)}{S + 1}, D_2 = D_1 \quad (3.3)$$



Obrázek 3.4: Grafické znázornění principu činnosti podvzorkovací vrstvy. Vstupní rozměry ($224 \times 224 \times 64$) jsou pomocí filtru o velikosti 2 a krokem 2 podvzorkovány na výstupní rozměry ($112 \times 112 \times 64$). Převzato z [15].

3.2.3 Plně propojená vrstva

Tato vrstva je běžně používána v běžných neuronových sítích. Neurony v plně propojené vrstvě jsou plně propojeny se všemi neurony v předešlé vrstvě a následující vrstvě. Jejich aktivace tedy může být vypočítána násobením matic. Samotné neurony náležící stejné vrstvě však nemají žádná propojení [15]. Princip propojení neuronů je znázorněn na obrázku 3.5.



Obrázek 3.5: Grafické znázornění plně propojené vrstvy. Vyobrazeny jsou čtyři vrstvy, kde neurony sousedních vrstev jsou plně propojeny, ale neurony náležící stejné vrstvě nejsou propojeny. Převzato z [15].

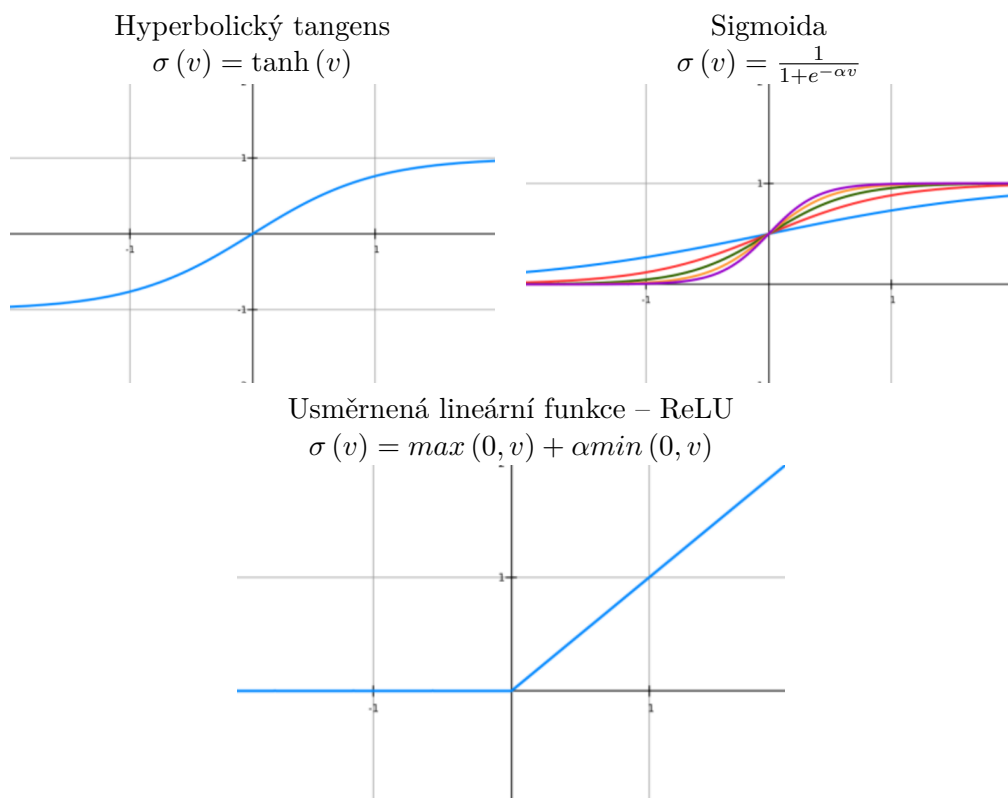
3.2.4 Aktivační funkce

Aktivační funkce provádí transformaci vnitřního potenciálu a vygenerování výstupu. Různé funkce mají různé vlastnosti. Některé jsou univerzální, jiné se hodí ke specifickým problémům. U aktivačních funkcí jsou žádané především dvě vlastnosti: diferencovatelnost a nelinearita. První zmíněná vyplývá z Backpropagation algoritmu. Kdyby se místo nelineární funkce použila funkce lineární, výsledkem by byla opět a jen lineární funkce. U neuronových sítí je však žádané transformovat vstupní prostor tak, aby byl lineárně oddělitelný [7].

Sigmoida se nejčastěji používá se střední hodnotou v bodě (0,0). Takové sigmoidy jsou liché funkce a díky svému tvaru výborně modelují výstup, protože jeho hodnota bude okolo nuly. K sigmoidě lze také přidat parametr α , který ovlivňuje zakřivení sigmoidy. Tento parametr lze učit a adaptovat.

Hyperbolický tangens má podobné vlastnosti jako sigmoida, ale u většiny klasifikačních úloh konverguje rychleji.

Usměrněná lineární funkce (ReLU, *Rectified linear unit*) není shora ohraničená a díky tomu umí síť odhalit míru chyby. Tato funkce je velmi často používaná.



Tabulka 3.1: Průběh a vzorce vybraných aktivačních funkcí. Převzato z [7].

3.2.5 Učení neuronové sítě

Problém učení neuronových sítí spočívá v minimalizaci funkce, která udává velikost chyby neuronové sítě nad danou množinou trénovacích dat. Tato funkce je nazývána chybová funkce (*loss function*) a jednou z nejběžněji používaných je střední kvadratická chyba (*MSE*, *Mean Squared Error*) definována následně [25]:

$$MSE := \frac{1}{n} \sum_{t=1}^n e_t^2 \quad (3.4)$$

kde n je zpravidla počet tříd a e je rozdíl mezi predikcí modelu neuronové sítě p a správnou třídou t , tedy $e = p - t$.

Pro vyhledání minima funkce je potřeba vypočítat její gradient, který udává směr růstu funkce a v opačném směru tedy funkce klesá. Po dostatečném počtu kroků ve směru klesání funkce lze dojít do minima, což značí úspěšné natrénování modelu neuronové sítě.

Výpočet gradientů je proveden pomocí algoritmu zpětného šíření chyby (*Error Back Propagation*). Nejprve je proveden dopředný průchod sítí, kdy se spočítají a uloží aktivace všech neuronů a poté se spočítají parciální derivace všech výstupních neuronů. Následně je proveden zpětný průchod sítí a jsou rekurzivně počítány parciální derivace všech neuronů v síti vždy z parciálních derivací neuronů v následující vrstvě. Nakonec se spočítají derivace vah a biasů z aktivací předešlé vrstvy a derivací následující vrstvy. Jednotlivé gradienty vah se od vah odečítají.

Pro učení neuronových sítí se využívá algoritmus Stochastic Gradient Descent. Je to iterační algoritmus, který pracuje nad podmnožinou dat z množiny trénovací datové sady. Průchod jedné této podmnožiny se nazývá iterací a spočívá ve výpočtu gradientů pomocí algoritmu zpětného šíření chyby. Pomocí získaných gradientů se poté v jednotlivých vrstvách upraví hodnoty vah a biasů následně:

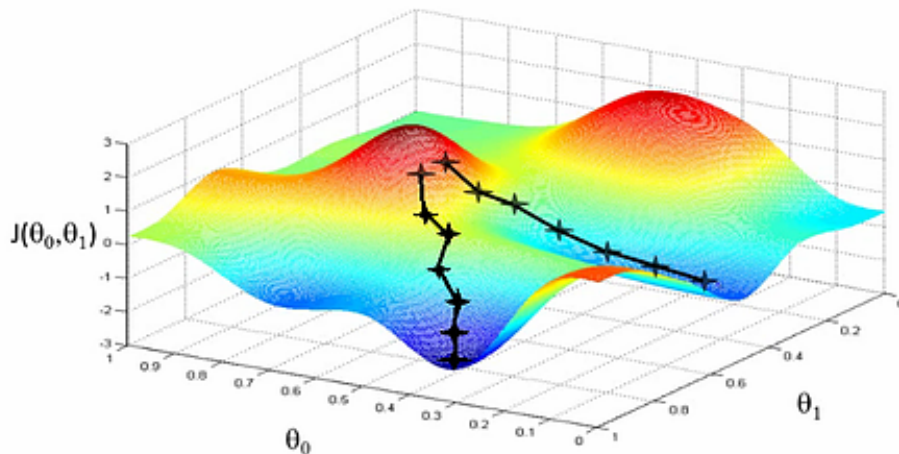
$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \quad (3.5)$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b) \quad (3.6)$$

kde l je vrstva, W je váha, b je bias, α je rychlost učení *learning rate*, která udává intenzitu aplikace vypočítaných změn parametrů a $J(W, b)$ je hodnota chybové funkce. Tím je proveden jeden krok ve směru klesání funkce a po dostatečném počtu kroků je dosaženo minima. U algoritmu Stochastic Gradient Descent může vzniknout problém takzvaného zaseknutí, kdy je možno dojít do lokálního minima a nikdy se nedostat ke globálnímu minimu. Tento problém je zachycen na obrázku 3.6, kde i při volbě mírně odlišného místa začátku výpočtu lze dojít do dvou odlišných lokálních minim [12].

3.3 Rekurentní neuronové sítě

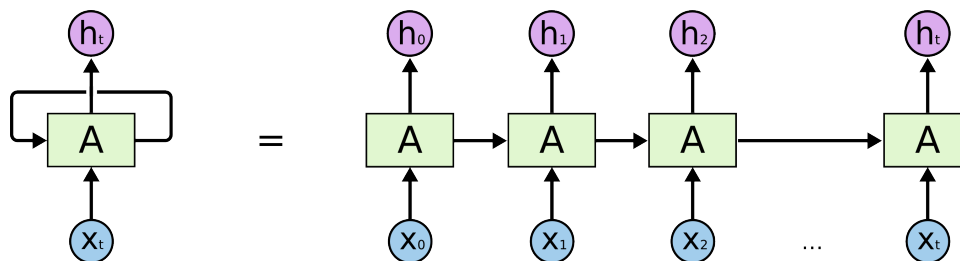
Tradiční neuronové sítě provádějí predikci pouze na základě jednoho vstupu, tedy pro stejný vstup bude vždy stejný výstup. Toto chování je velkým nedostatkem v případě, kdy je potřeba získat informace na základě spojených vstupních sekvencí. Rekurentní neuronové sítě řeší tento problém zavedením vnitřních smyček, které propojují předchozí rozhodnutí sítě. Toto je často označováno jako skrytý vnitřní stav či vnitřní paměť. Tedy rekurentní



Obrázek 3.6: Vizualizace průběhu algoritmu Stochastic Gradient Descent a jeho problému takzvaného zaseknutí. Model zachycené neuronové sítě obsahuje dva parametry a povrch v prostoru parametrů udává, jakou chybu bude tato neuronová síť mít při volbě daných vah sítě. Převzato z [12].

neuronové sítě mají dva vstupy – současný, stejně jako klasické neuronové sítě a předchozí. Z toho vyplývá, že predikce rekurentní neuronové sítě v kroku t je závislá na predikci v kroku $t-1$, tedy zachování kontextu vstupní sekvence [8]. Vnitřní smyčky si lze představit jako několik kopií stejné sítě, kde při každém kroku je informace poskytována následníkovi, což je znázorněno na obrázku 3.7.

Problémem klasických rekurentních sítí je schopnost naučit se pouze sekvence s krátkým časovým krokem, tedy výše ilustrovaný případ kroků $t-1$ a t jsou se schopny naučit výrazně lépe, zatímco například rozestup kroků $t-100$ a t podstatně hůře. Tento problém řeší síť složené z jednotek LSTM.



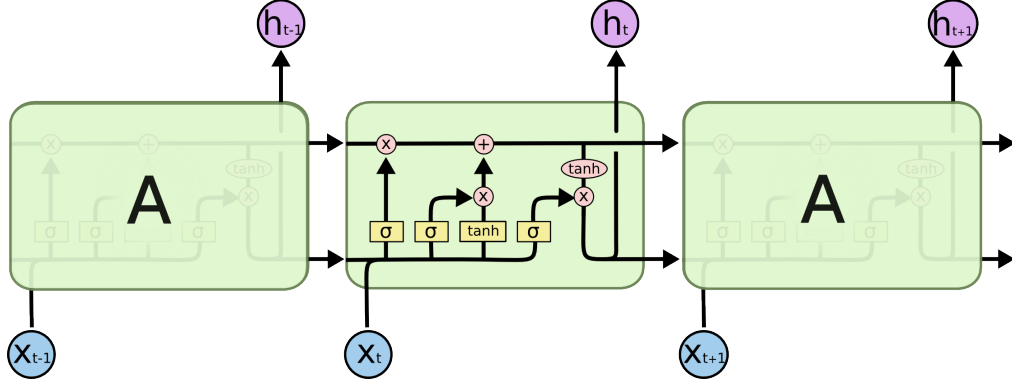
Obrázek 3.7: Znázornění rekurentní neuronové sítě a rozložení vnitřní smyčky do několika kopií sítě. Převzato z [18].

3.3.1 Jednotka LSTM

Neuronové sítě složené z LSTM jednotek (*Long Short-Term Memory Unit*) jsou speciální typy rekurentních neuronových sítí, které jsou schopny se naučit závislosti s delšími časo-

vými rozestupy. Poprvé je představili pánové Hochreiter a Schmidhuber [11] a popisuje je Christopher Olah [18].

Vnitřní struktura LSTM jednotky je zachycena na obrázku 3.8. Lze vidět, že hodnota výstupního vektoru \mathbf{h} v čase t je závislá na hodnotě v čase $t-1$ a dále je využita v čase $t+1$. Klíčovou myšlenkou za LSTM jednotkou je vektor \mathbf{C} , který prochází horizontálně skrz tuto jednotku (obrázek 3.9 vlevo) a uchovává stav buňky. Jednotka LSTM má pak možnost přidávat či odebírat uchovanou informaci pomocí vnitřní struktury hradel. Jednotlivá hradla mohou nabývat hodnoty v intervalu $(0, 1)$, kdy v hodnotě 0 není propuštěna žádná informace, zatímco naopak v hodnotě 1 je propuštěno vše.



Obrázek 3.8: Obrázek zachycuje zřetězení tří LSTM jednotek a náhled na jejich vnitřní strukturu. Převzato z [18].

Prvním hradlem je tzv. *forget gate* (obrázek 3.9 vpravo), což je vrstva s aktivační funkcí sigmoida (viz. 3.2.4). Jeho úkolem je rozhodnout, které informace ve vektoru \mathbf{C} zůstanou a které se naopak odeberou. Vstupem hradla je \mathbf{h}_{t-1} a \mathbf{x}_t , tedy výstupní vektor předchozího kroku a vstupní vektor aktuálního kroku. Výstupem je vektor \mathbf{f}_t , který se počítá dle vzorce (3.7), kde \mathbf{W}_f jsou váhy vrstvy a b_f je bias vrstvy.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + b_f) \quad (3.7)$$

Dalším hradlem je tzv. *input gate* (obrázek 3.10 vlevo), jehož úkolem je určit, které nové informace budou uloženy do stavu buňky. Toto hradlo se sestává ze dvou vrstev. Vstupem obou vrstev jsou opět vektory \mathbf{h}_{t-1} a \mathbf{x}_t . První vrstva obsahuje aktivační funkci sigmoida a jejím úkolem je určit, které hodnoty se budou aktualizovat. Výstupní vektor \mathbf{i}_t je vypočítán dle vzorce (3.8), kde \mathbf{W}_i jsou váhy vrstvy a b_i je bias vrstvy. Druhá vrstva má aktivační funkci hyperbolický tangens (viz. 3.2.4) a konstruuje vektor $\tilde{\mathbf{C}}_t$ obsahující nové kandidátní hodnoty, který se počítá pomocí vzorce (3.9), kde \mathbf{W}_C jsou váhy vrstvy a b_C je bias vrstvy.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + b_i) \quad (3.8)$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + b_C) \quad (3.9)$$

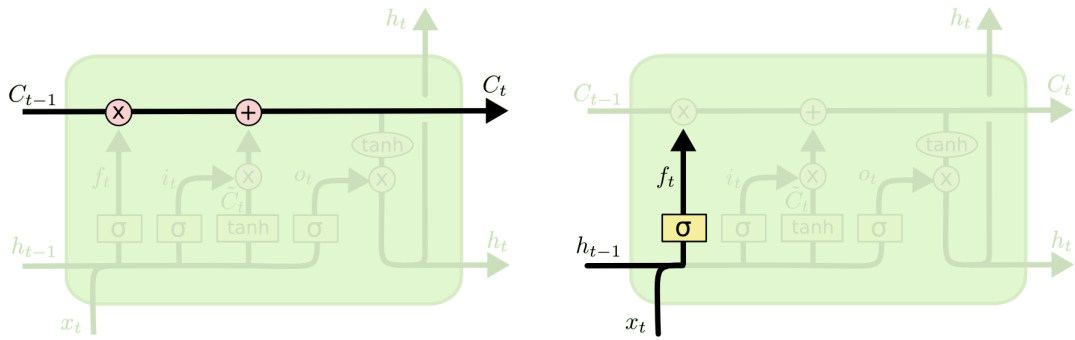
Následujícím krokem je vlastní aktualizace staré hodnoty stavu buňky \mathbf{C}_{t-1} , čímž se získá nový stav buňky \mathbf{C}_t . Aktualizace se provádí pomocí vzorce (3.10) a spočívá v násobení staré hodnoty buňky \mathbf{C}_{t-1} s vektorem \mathbf{f}_t , čímž se odstraní nechtěné hodnoty. Poté se přičte výsledek násobení \mathbf{i}_t a $\tilde{\mathbf{C}}_t$, což má za následek přidání nových hodnot.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.10)$$

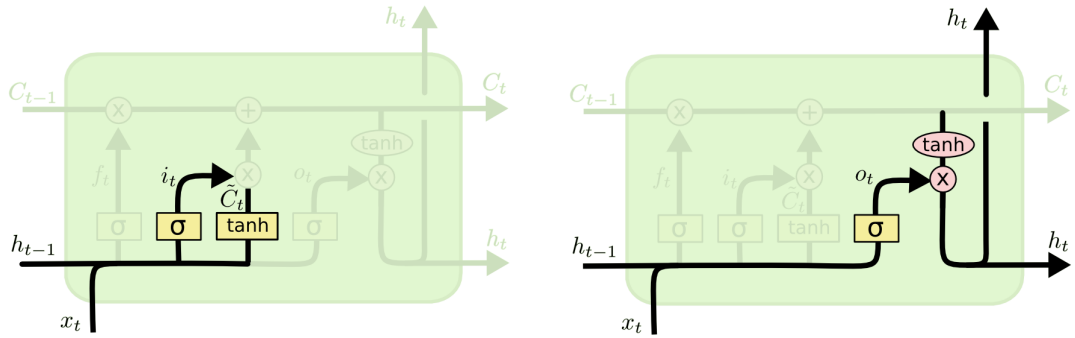
Nakonec je potřeba vypočítat výstup h_t , což je znázorněno na obrázku 3.10 vpravo. Hodnota výstupu je založena na vyfiltrovaném stavu buňky C_t . Nejprve je použita vrstva s aktivační funkcí sigmoida, která vybere části stavu buňky, které budou na výstupu. Toto se vypočítá pomocí vzorce (3.11). Následně se pomocí vrstvy s aktivační funkcí hyperbolický tangens posune stav buňky C_t do intervalu $(-1, 1)$ a vynásobí se s výstupem vrstvy s sigmoidou (vzorec (3.12), čímž se vybere pouze část původních hodnot stavu buňky.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.11)$$

$$h_t = o_t * \tanh(C_t) \quad (3.12)$$



Obrázek 3.9: Vlevo vektor C , který uchovává stav buňky. Vpravo hradlo *forget gate* jehož výstupní vektor f_t je vypočítán dle vzorce (3.7). Převzato z [18].



Obrázek 3.10: Vlevo hradlo *input gate*, jehož výstup obsahuje nové hodnoty pro přidání do stavu buňky. Vpravo hradlo *output gate*, které vytváří výstup jednotky LSTM pomocí výběru hodnot ze stavu buňky. Převzato z [18].

Kapitola 4

Návrh systému

V této kapitole je popsán návrh systému pro odhad rychlosti vozidla z palubní kamery. Výsledný systém by měl být schopen odhadnout rychlost vozidla za libovolných světelných podmínek, počasí či různých ročních období. Jediným vstupem systému je obrazová informace zachycená palubní kamerou s předpokladem dobrého výhledu ve směru jízdy vozidla.

V první části této kapitoly uvádím dekompozici systému na jednotlivé nezávislé celky. V druhé a zároveň poslední sekci této kapitoly rozeberu omezení navrženého systému vzhledem k vybraným metodám.

4.1 Dekompozice systému

Systém pro odhad rychlosti vozidla jsem rozdělil na tři části znázorněné na obrázku 4.1. První z nich se zabývá předzpracováním vstupních snímků. Jejím úkolem je připravit snímek do podoby, kterou očekává další část v pořadí. Tou je konvoluční neuronová síť, která má za úkol samotný odhad rychlosti. Poslední částí systému je postprocessing, který na výstupu konvoluční neuronové sítě provede finální úpravy.



Obrázek 4.1: Návrh systému pro odhad rychlosti vozidla ze záznamu palubní kamery.

4.1.1 Preprocessing

Prvním krokem systému je preprocessing. Vstupem tohoto modulu jsou „surové“ snímky zachycené palubní kamerou. Tyto snímky obsahují řadu informací, které jsou zanedbatelné či přímo zbytečné. Příkladem může být palubní deska či kapota vozidla nebo neměnná obloha. Tyto regiony obrazu nenesou žádnou důležitou informaci potřebnou pro odhad rychlosti a proto je možné je odstranit. Dalším regionem, na který je potřeba se zaměřit, je střed snímku, kde ve valné většině případů bude zachyceno jiné vozidlo. Tato část snímku tedy ve velkém počtu případů nebude obsahovat informace potřebné pro odhad rychlosti či dokonce bude obsahovat chybné informace vzhledem k pohybu druhého vozidla. Takto připravený snímek je potřeba transformovat na tvar, který očekává další modul v pořadí – konvoluční neuronová síť.

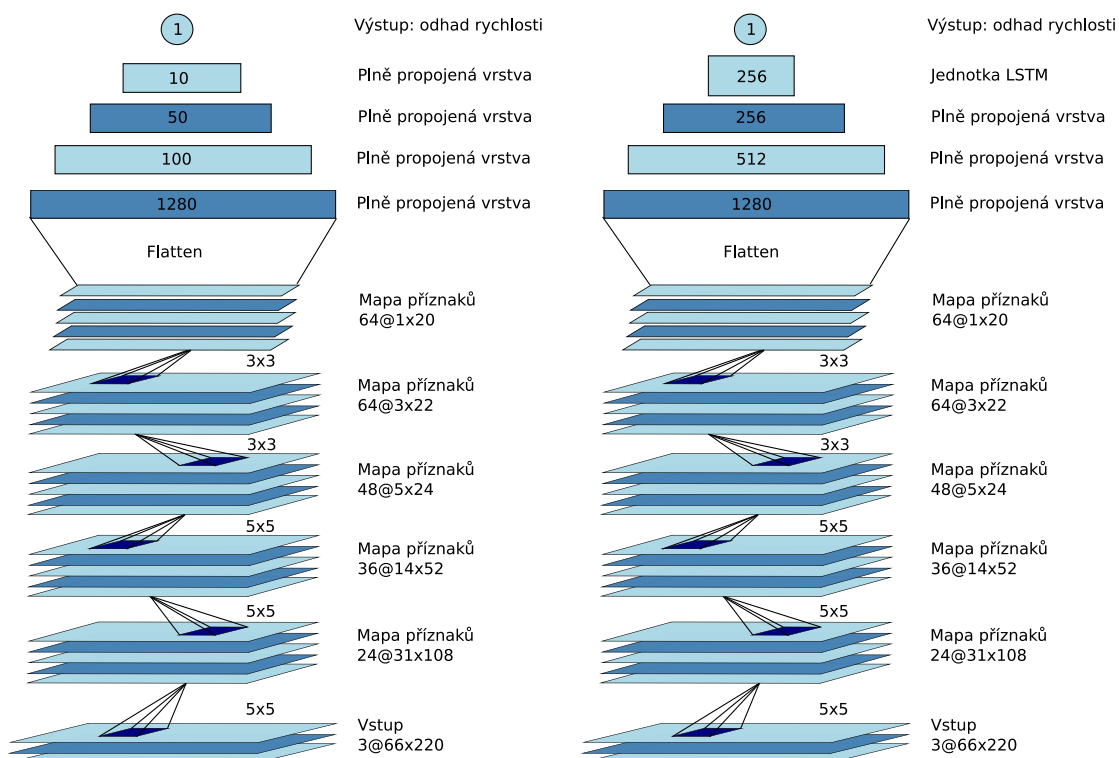
Jednou ze zvolených metod je odhad optického toku. Tato metoda potřebuje dva po sobě jdoucí snímky záznamu. Po zpracování každé dvojice snímků je proveden odhad optického toku, jenž je vložen na vstup následujícího modulu.



Obrázek 4.2: Předzpracování vstupního snímku.

4.1.2 Konvoluční neuronová síť

Vstupem konvoluční neuronové sítě je snímek obsahující optický tok mezi dvěma po sobě následujícími snímky. Na základě různých hodnot vektorů optického toku je potřeba přiřadit správnou třídu rychlosti dané dvojici snímků. Pro tento účel jsem použil konvoluční neuronovou síť převzatou z [4], která původně sloužila pro řízení vozidla na základě snímané cesty. Architektura sítě je znázorněna na obrázku 4.3 vlevo. V pravé polovině obrázku je pak modifikovaná architektura sítě s použitím jednotky LSTM.

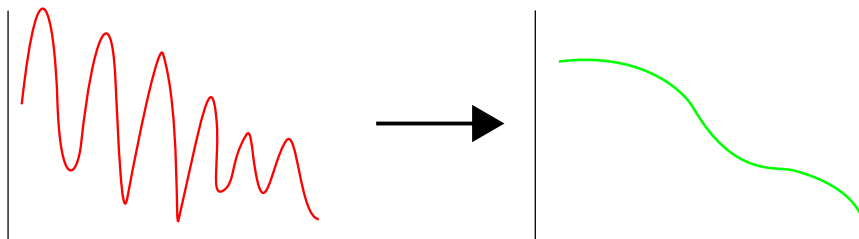


Obrázek 4.3: Architektura konvoluční neuronové sítě použité pro odhad rychlosti vozidla. Vlevo bez jednotky LSTM, vpravo s použitím jednotky LSTM. Převzato z [4].

4.1.3 Postprocessing

Posledním krokem systému je postprocessing, jehož úkolem je vylepšit odhad rychlosti provedený konvoluční neuronovou sítí. Výstup sítě může být zašumělý nebo obsahovat náhlé výkyvy hodnot, které jsou způsobené špatným odhadem optického toku či chybným odhadem rychlosti. Odstranění těchto chybných hodnot je provedeno pomocí průměrování posuvným oknem.

Alternativou či doplňkem k tomuto postupu může být rekurentní neuronová síť vložená na výstup předchozího modulu. Tato síť by měla ošetřit náhlé výkyvy odhadu rychlosti.



Obrázek 4.4: Postprocessing odhadnuté rychlosti.

4.2 Omezení systému

Na úvod je vhodné uvést, že předpokládám správné umístění palubní kamery, tedy za čelním sklem vozidla (či podobné umístění) s optickou osou souběžně se směrem jízdy vozidla a dobrým výhledem na dopravní situaci před vozidlem. Omezení navrženého systému pak vyplývá především z použitých metod.

Mezi největší problémy optického toku patří částečné či úplné zakrytí snímané scény, ať již úmyslné, příkladem může být umístění předmětu před objektiv kamery, či vzniklé v důsledku dopravní situace. Druhé uvedené může být způsobeno blízkým kolmým průjezdem jiného vozidla, především větších rozměrů. V tomto případě dojde k odhadu optického toku pro projíždějící vozidlo a tedy dojde k výpočtu chybné hodnoty pro využití při odhadu rychlosti vozidla. Dalším problémem je nedostatek výrazných bodů pro sledování a z toho vyplývající nemožnost odhadnout optický tok. Tato situace může nastat v rozlehlých krajinách s minimální či žádnou strukturou povrchu okolního terénu.

Vzhledem k monotónnosti trénovacích dat vzhledem ke globálnímu měřítku, lze předpokládat, že konvoluční neuronová síť bude i přes řádnou generalizaci odhadovat chybné hodnoty rychlosti. Toto bude platit především pro prostředí značně odlišné od lokality pořízení trénovací datové sady. Lze tedy říci, že odhad rychlosti například v pouštním prostředí bude zatížen značnou chybou. Menší chybu odhadu rychlosti může také způsobit některý z dalších parametrů jako je změna kamery, umístění kamery, roční období a jiné.

Kapitola 5

Implementace systému

Tato kapitola popisuje implementaci systému pro odhad rychlosti vozidla ze záznamu palubní kamery a technologie a data k tomu potřebná. Nejprve je popsána datová sada, která byla vytvořena pro účely implementace a testování systému. Dále je zde popsána samotná implementace navrženého systému a představeny výsledky této implementace. Závěrem kapitoly jsou stručně představeny technologie, které byly použity při implementaci systému.

5.1 Vytvořená datová sada

Pro úspěšné naučení konvoluční neuronové sítě je potřeba dostatečně velká datová sada. Tato sada by měla obsahovat co nejrozmanitější data, aby výsledná neuronová síť dosahovala dobrých výsledků nad neznámými daty. Součástí sady by také měla být kvalitní anotace dat, neboť v opačném případě by došlo k zavedení chyby již při trénování sítě.

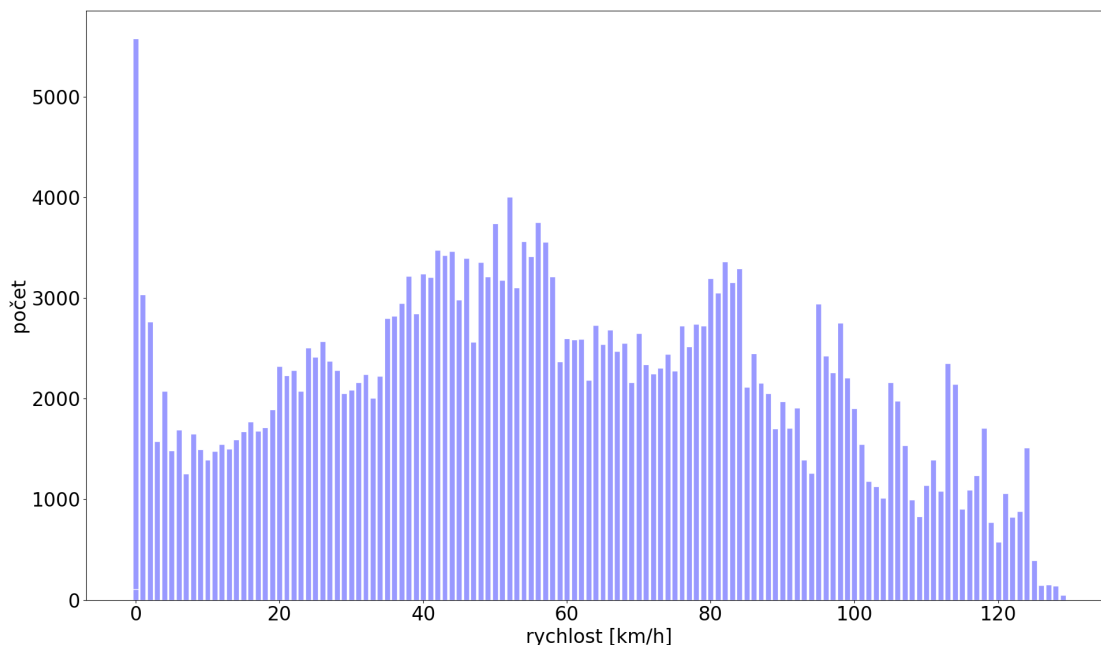
Z tohoto důvodu byla v rámci této práce vytvořena nová datová sada, která zachycuje pohled z palubní kamery osobního vozidla. Palubní kamera je umístěna na čelním skle vozidla souběžně se směrem jízdy tak, aby byl zajištěn dobrý výhled na dění před vozidlem. Data byla sbírána na území města Brna a v jeho okolí. Několik video segmentů také zachycuje cestu po dálnici do jiných regionů. Pořízení datové sady probíhalo v období podzimu až začátku zimy roku 2017. Zaznamenávacím zařízením je palubní kamera TrueCam A4¹ s přídatným GPS modulem² pro zaznamenávání informací o jízdě. Celkem bylo nasbíráno zhruba 22 hodin záznamu a obrázek 5.2 zachycuje několik náhledů na vytvořená videa.

5.1.1 Rozdělení datové sady

Vzhledem ke zvoleným metodám strojového učení je potřeba vytvořenou datovou sadu rozdělit na trénovací, validační a testovou sadu. Sada pro trénování konvoluční neuronové sítě odpovídá zhruba 70% celkové velikosti, validační sada pak zhruba 20% a testovací sada zbývajícím 10%. Jednotlivé sady jsou ručně sestaveny tak, aby obsahovali co nejvyrovnanější poměr jednotlivých rychlostí. Výjimkou je rychlost „0“, která je v datové sadě hojně zastoupena díky četnému stání vozidla v městském prostředí. Obrázek 5.1 udává zastoupení jednotlivých hodnot rychlostí v rámci trénovací datové sady. Jak lze vidět, v sadě se častěji vyskytují hodnoty, které jsou blízké zákonným limitům na pozemních komunikacích. Také lze vidět, že v sadě jemně převažuje záznam z nižších rychlostí v porovnání se záznamem z dálnice.

¹<http://www.truecam.cz/kamery-do-auta-truecam/truecam-a4/>

²<http://truecamshop.cz/cs/truecam/187-truecam-gps-s-detekci-radaru-2016-8594175350579.html>



Obrázek 5.1: Znázornění rozložení jednotlivých hodnot rychlosti zastoupených v trénovací datové sadě.

5.1.2 Zpracování datové sady

Ke zpracování „surových“ video záznamů kamery jsem použil program DATAKAM Player³ (internetová stránka produktu ani společnosti, která jej vyvinula, již není dostupná a tak odkazuji alespoň video tutoriál, který uvádí výrobce palubní kamery). Použitá palubní kamera ukládá nahraný video záznam po dílčích pěti minutových segmentech při snímkovací frekvenci třicet snímků za sekundu. Prvním krokem zpracování datové sady je tedy sloučení jednotlivých segmentů do ucelených částí. Zaznamenané informace o jízdě z GPS modulu jsou ukládány přímo v kontejneru videa. Dalším krokem je tedy extrakce těchto záznamů. O zaznamenané jízdě je uchovávána celá řada informací: GPS souřadnice, azimut, nadmořská výška, rychlost, čas a uražená vzdálenost. Tyto informace jsou pomocí výše uvedeného programu exportovány a uloženy ve formátu CSV (*Comma-separated values*) a to v následující podobě:

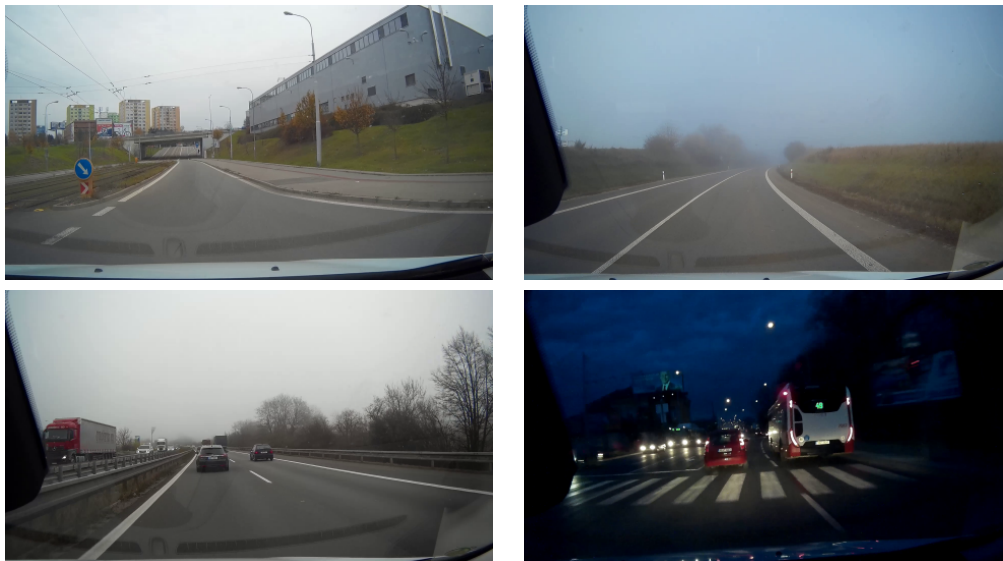
| Datum; | Čas; | X; | Y; | Rychlost; | Azimut; | Délka cesty; |
|------------|-----------|----------|----------|-----------|---------|--------------|
| 3.11.2017; | 11:46:12; | 16.6582; | 49.1899; | 72; | 308; | 3027; |

kde X a Y jsou GPS souřadnice, rychlost je uváděna v kilometrech za hodinu a délka cesty v celých ujetých metrech. Tyto hodnoty jsou zaznamenávány jednou za sekundu.

Zpracování trénovací a validační datové sady

Další zpracování spočívá v převodu snímkovací frekvence ze třiceti snímků za sekundu na pět snímků za sekundu. Tento převod je implementován prostým „zahazováním“ pěti snímků a ponecháním právě šestého snímku. Tento snímek je oříznut o okrajové části, které nenesou

³<https://www.youtube.com/watch?v=ZvRBuEjNGz4&feature=youtu.be>



Obrázek 5.2: Ukázka snímků z vytvořené datové sady.

žádnou užitečnou informaci a dále jsou změněny jeho rozměry na požadovaný vstup konvoluční neuronové sítě. Poté je snímek uložen jako obrázek a to z důvodu lepší manipulace při učení neuronové sítě a dále je zapsána cesta jeho současného umístění do anotačního souboru.

Pro jednotlivé takto zpracované snímky je nyní potřeba zpracovat soubor se záznamem o jízdě. V rámci této práce je zájmovým údajem především naměřená rychlost. Vzhledem k tomu, že údaje jsou zapisovány jednou za sekundu, zatímco jednotlivých obrázků existuje pět za sekundu, je potřeba tyto údaje doplnit. Za tímto účelem byl vytvořen program, který čte zaznamenané hodnoty a dle požadované snímkové frekvence videa lineárně doplní údaje o rychlosti v rámci jedné sekundy. Tento údaj je poté zapsán do anotačního souboru k odpovídajícímu snímku. Výstupem je tedy sada snímků a anotační soubor obsahující cestu ke snímku a přiřazenou odpovídající rychlost, v následujícím formátu:

| Cesta | Rychlost |
|----------------------------|----------|
| ../../../../data/90554.png | 43 |

kde rychlost je uvedena v kilometrech za hodinu.

Nepříliš častým jevem v souboru záznamů o jízdě je chybějící hodnota na několika po sobě následujících řádcích. Toto je způsobeno krátkodobou ztrátou signálu například při průjezdu tunelem či změnou družic. Tento jev je z dat odstraněn dopočítáním chybějících hodnot. Výpočet spočívá v nalezení poslední zaznamenané rychlosti a následující zaznamenané rychlosti a dále lineárním rozložením změny v rychlosti na úsek s chybějícími daty. Toto řešení předpokládá pouze krátkodobý výpadek signálu a je chybné v případě, kdy úsek obsahuje například zastavení vozidla a následný rozjezd. Takovouto situaci však datová sada neobsahuje.

Zpracování testovací datové sady

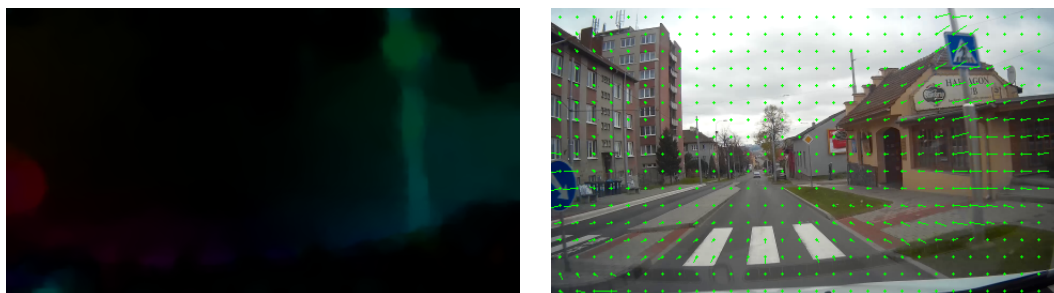
Testovací datová sada je ponechána ve výše popsaném formátu, tedy samostatný video záznam a anotační soubor obsahující informace o jízdě. Z přidělené části datové sady je však

vytvořeno několik krátkých úseků o délce třicet sekund. Tyto úseky jsou určeny pro testování natrénované konvoluční neuronové sítě a proto obsahují vybrané zajímavé události. Zejména různorodou hodnotu rychlosti, konstantní rychlost, akceleraci, deceleraci, městské prostředí, venkovské prostředí, dálniční prostředí a další. Dále jsou vytvořeny dva delší úseky, kdy jeden obsahuje standardní cestu městem a druhý standardní cestu po dálnici a silnici I. třídy. Tyto úseky by měly reprezentovat běžný záznam jízdy.

5.2 Optický tok

Pro výpočet optického toku je použita třída `DISOpticalFlow`⁴. Tato třída je implementací algoritmu Dense Inverse Search pro výpočet optického toku, který je rozebrán v sekci 3.1, v knihovně OpenCV. Nejprve je potřeba vytvořit instanci třídy `DISOpticalFlow` a to pomocí metody `createOptFlow_DIS()`. Tato metoda nabízí výběr mezi třemi úrovněmi nastavení, jež spočívá buď ve vyšší rychlosti a horší kvalitě optického toku či naopak. Po vytvoření instance je již možné provádět jednotlivé výpočty optického toku a to pomocí metody `calc()`. Vstupními parametry této metody jsou dva snímky převedené do odstínů šedi, mezi kterými je poté vypočítán optický tok. Výstupem metody je tedy dvourozměrné pole, které obsahuje silovou a směrovou složku optického toku. Ilustrace odhadnutého optického toku je na obrázku 5.3.

Převedení do barevného modelu HSV spočívá v převedení úhlu směru pohybu do polárního systému, tedy z radiánů na stupně. Na základě vypočítaného úhlu se provede přiřazení barvy z barevného kola a tedy přiřazení do složky barevného modelu HSV – barevný tón (*Hue*). Druhým krokem je normalizace hodnoty síly pohybu do rozmezí (0–255). Normalizovaná hodnota je poté přiřazena do složky barevného modelu HSV – hodnota jasu (*Value*). Složka udávající sytost barvy (*Saturation*) je nastavena na maximální hodnotu, tedy 255.



Obrázek 5.3: Grafické znázornění optického toku mezi dvěma snímky. Vlevo převedením do barevného modelu HSV, vpravo znázornění vektorů pohybu ve vybraných místech.

5.3 Trénování konvoluční neuronové sítě

Trénování konvoluční neuronové sítě probíhá nad trénovací a validační datovou sadou popsanou v sekci 5.1.2. Vzhledem k velikosti trénovací sady je potřeba využít tzv. generátoru. Generátor je funkce, která dává postupně k dispozici hodnoty a definuje se pomocí klíčového slova `yield`. Tedy každá funkce, která obsahuje klíčové slovo `yield` je generátorová funkce (*generator function*). Voláním metody `next()` se pak provede funkce až do místa

⁴https://docs.opencv.org/3.3.1/da/d06/classcv_1_1_optflow_1_1_DISOpticalFlow.html

klíčového slova `yield` a tam se zastaví. Opětovným voláním metody `next()` se funkce začne vykonávat od předchozího místa zastavení a poté se opět zastaví za klíčovým slovem `yield`. Jednoduchý příklad použití generátoru a jeho výstupu je uveden níže:

```
def Generator():
    i = 0
    while True:
        i = i + 1
        yield i

gen = Generator()
print(next(gen))
print(next(gen))
```

```
>>> 1
>>> 2
```

Pomocí anotačního souboru jsou vytvořeny všechny kombinace dvojic dvou po sobě následujících snímků a odpovídajících rychlostí. Tyto informace jsou pak vloženy na vstup generátoru, který načte snímky z uvedených cest a vypočítá průměr z uvedených rychlostí. Mezi dvojicí snímků je poté vypočítán optický tok (viz. sekce 5.2). Takto zpracovaná data jsou poté spojována do větších celků – dávek a následně poskytnuty k dispozici pomocí klíčového slova `yield`.

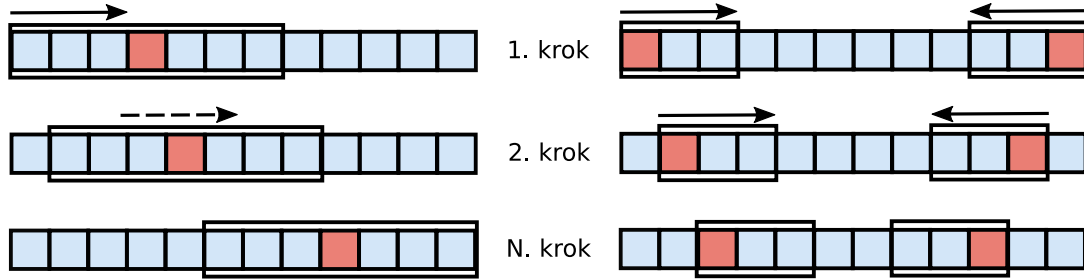
Velikost dávky udává četnost a celkový počet úprav hodnot vah a biasů v jednotlivých vrstvách konvoluční neuronové sítě a je volena od 32 až po 8 snímků optického toku v závislosti na průběhu trénování. Koeficient rychlosti učení (*learning rate*) se postupně snižuje z hodnoty 10^{-3} na hodnotu 10^{-6} v závislosti na počtu provedených průchodů nad trénovacími daty. Validační data jsou zpracována obdobným způsobem.

5.4 Odhad rychlosti

Samotný odhad rychlosti probíhá nad předem nezpracovaným videem. Po načtení videa je získána jeho snímkovací frekvence a jsou načítány jednotlivé snímky. Tato implementace pracuje se snímkovací frekvencí pět snímků za vteřinu a tedy je dále propouštěn jen každý N -tý snímek tak, aby došlo ke správnému převodu ze zdrojové snímkové frekvence. Snímek je dále oříznut o okrajové části, které nenesou žádnou důležitou informaci. Poté je změněna jeho velikost na 220×66 , což je požadovaný vstupní formát konvoluční neuronové sítě. Počínaje druhým snímkem videa je vždy pro současný a předchozí snímek vypočten optický tok tak, jak je popsáno v sekci 5.2. Současný snímek je poté uchován pro nadcházející cyklus a optický tok je vložen na vstup konvoluční neuronové sítě.

Odhad rychlosti nad jednotlivými dvojicemi snímků je uchováván v poli, jehož délka odpovídá počtu snímků. Jakmile je proveden odhad rychlosti nad celým video záznamem, je proveden volitelný postprocessing, jehož úkolem je vyhlazení odhadnutých dat (viz. sekce 4.1.3). Postprocessing je implementován jako okno o velikosti N , které se postupně posouvá hodnotu po hodnotě ke konci sekvence. Aby se zabránilo většímu zkreslování dat, tak se aktuálně zpracovávaná položka nachází na pozici $N / 2$ posuvného okna. Samotné zpracování spočívá v sečtení hodnot v rozsahu okna a následném vydělení velikostí okna, tedy spočítání průměru. Výstupem je pak pole obsahující vyhlazený průběh rychlostí v rámci zkoumaného video záznamu.

Vzhledem k tomu, že posuvné okno pracuje s položkou nacházející se na prostřední pozici, je možné tento přístup aplikovat až od položky pole na pozici $N \div 2$ a ukončit na pozici $L - N \div 2$, kde L je délka daného pole. Tento problém je vyřešen pomocí okna o velikosti $N \div 2$, jehož aktuální položka se nachází na nejlevější respektive na nejpravější pozici. Výpočet průměru je proveden stejným způsobem a okno se posouvá směrem ke konci pole respektive směrem k začátku pole. Tento přístup zavádí drobné zkreslení výsledků v okrajových hodnotách pole. Princip činnosti posuvného okna je znázorněn na obrázku 5.4.



Obrázek 5.4: Princip činnosti posuvného okna použitého v postprocessingu. Vlevo je hlavní okno pracující nad prostřední hodnotou, vpravo okna řešící okrajové hodnoty. Vyobrazen je první, druhý a poslední krok.

5.5 Validace natrénované neuronové sítě

Posledním krokem v rámci implementace je ověření natrénované konvoluční neuronové sítě. Ověření je nutné, neboť síť se mohla naučit rozpoznávat rychlost vozidla pouze nad daty obsaženými v trénovací a validační datové sadě. Z tohoto důvodu je síť otestována nad testovací datovou sadou, která je popsána v sekci 5.1.2. Konkrétně nad dvěma delšími úseky zachycujícími záznam běžné jízdy.

Po provedení odhadu rychlosti, jak je popsáno v sekci 5.4, je porovnána naměřená rychlost s rychlostí odhadnutou. Naměřená rychlost je součástí anotačního souboru testovací datové sady, který není předem zpracován. Z tohoto důvodu je potřeba doplnit chybějící údaje o rychlosti, které jsou způsobeny ztrátou signálu GPS modulu či zaznamenávací frekvencí jednou za sekundu. V obou případech je postup obdobný a to takový, že se vyhledá poslední a následující zaznamenaná rychlost. Poté jsou chybějící hodnoty lineárně doplněny vzhledem k rozdílu rychlostí.

Naměřená i odhadnutá rychlost jsou poté vyneseny do grafu spolu s dalšími metrikami, které jsou popsány dále v sekci 6.4.1. Tyto grafy jsou uvedeny na obrázku 5.5, kde sloupec odpovídá jedné video sekvenci. První video sekvence je záznam z městského prostředí, zatímco druhý záznam je pořízen částečně na dálnici a částečně na silnicích I. či II. třídy.

Jak lze vidět na obrázku 5.5 natrénovaná konvoluční neuronová síť dokáže odhadnout rychlost i nad předem neviděnými daty. Především lze říci, že neuronová síť dokáže reagovat na místa se značnou změnou rychlosti. Odhad rychlosti ve video sekvenci zachycující městské prostředí bez použití postprocessingu je příliš rozkolísaný, což je způsobeno tím, že odhad se provádí nad jednotlivými snímky bez znalosti většího kontextu. Při použití postprocessingu, tedy zprůměrování odhadnutých hodnot, je již odhad poměrně přesný. Na video sekvenci obsahující záznam z vyšších rychlostí jsou již výsledky znatelně horší i při použití postprocessingu. Při pohledu na kumulativní chybu lze však vidět, že je u obou video

sekvencí zhruba stejná. Tedy procentuální rozdíl naměřené a odhadnuté rychlosti se drží na stejných hodnotách u obou případů.

Z uvedených grafů vyplývá, že navržená a natrénovaná konvoluční neuronová síť dokáže reagovat na změny v optickém toku mezi dvěma snímky a s určitou chybou i odhadnout rychlost vozidla.

5.6 Použité technologie

Systém je implementován v programovacím jazyce Python s využitím knihoven OpenCV, Keras a TensorFlow, které jsou dále popsány. Výsledná aplikace je multiplatformní, ovládaná pomocí příkazového řádku s volitelným grafickým výstupem.

OpenCV

OpenCV⁵ (Open Source Computer Vision Library) je multiplatformní knihovna zaměřená na zpracování obrazu, počítačové vidění a strojové učení. Původně byla vyvíjena firmou Intel a v současnosti je volně šiřitelná pod licencí BSD. Knihovna je napsána v programovacím jazyce C++ a je možné ji využít v řadě dalších programovacích jazycích, např. Python, Java či C#. V této práci je použita ve verzi 3.3.

Keras

Keras⁶ je vysoko úroňová knihovna pro neurální sítě a může běžet nad knihovnami TensorFlow, Theano nebo CNTK. Knihovna je napsána v programovacím jazyce Python se zámerem rychlého prototypování a experimentování a je volně šiřitelná pod licencí MIT.

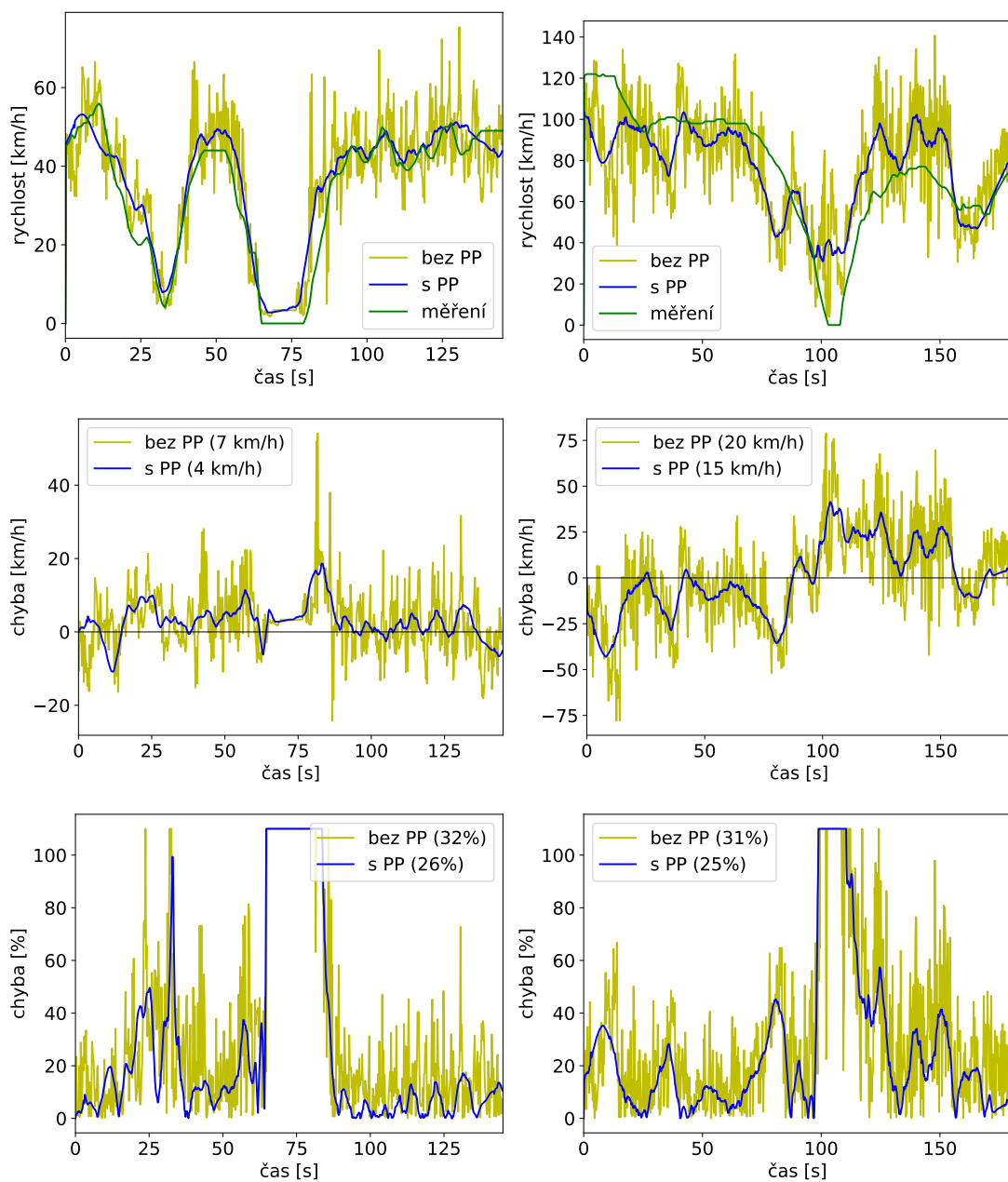
TensorFlow

TensorFlow⁷ je softwarová knihovna pro numerické výpočty. Původně byla vyvinuta týmem Google Brain pro strojové učení a hluboké konvoluční neuronové sítě, ale lze ji použít v celé řadě dalších problémů. Knihovna je volně šiřitelná pod licencí Apache 2.0.

⁵<https://opencv.org/>

⁶<https://keras.io/>

⁷<https://www.tensorflow.org/>



Obrázek 5.5: Odhadnutá rychlost na dvou sekvencích testovacích videí, kde PP je zkratka pro postprocessing. Vlevo záznam nižších rychlostí z městského prostředí a vpravo záznam vyšších rychlostí z dálnice a silnice I. a II. třídy.

Kapitola 6

Experimenty a vyhodnocení systému

V této kapitole se věnuji popisu provedených experimentů a následně jejich vyhodnocení. Experimenty byly provedeny za účelem vylepšení samotného odhadu rychlosti vozidla.

Experimentoval jsem s architekturou navržené konvoluční neuronové sítě, jejím učením a vstupem. Dále rozšířením neuronové sítě o rekurentní část v podobě LSTM jednotky (viz. sekce 3.3.1, jejíž účelem je zavedení kontextu v rámci určité délky sekvence snímků. Tedy v rámci dvou po sobě následujících snímků by nemělo docházet k velkým změnám hodnot odhadnuté rychlosti vozidla.

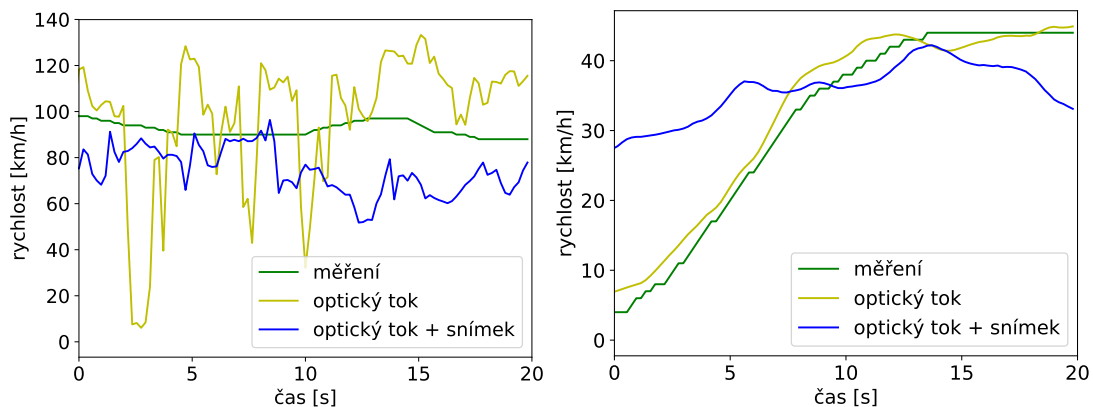
Vzhledem ke složitosti vstupních dat z palubní kamery vozidla jsem provedl několik experimentů věnujících se odfiltrování nežádoucích částí snímku či nežádoucího pohybu. Do těchto kategorií spadá například ostatní automobilová doprava či jiné pohybující se objekty, jež značně zkreslují odhad optického toku nebo pohyb kamery ve vertikálním směru, který je způsoben například akcelerací či přejetím nerovnosti na vozovce.

Na závěr této kapitoly jsou detailně uvedeny výsledky této práce. Dále jsou tyto výsledky rozebrány a popsány situace či úseky, kde dochází k největším chybám v odhadu rychlosti. Jako poslední je diskutován další možný vývoj této práce na základě získaných výsledků.

6.1 Vstup konvoluční neuronové sítě

Konvoluční neuronová síť, jež je popsána v kapitole 4 – Návrh systému, má na vstupu pouze matici s odhadem optického toku. Samotný optický tok může být v „surovém“ formátu, tedy dvě složky – síla a směr pohybu, nebo je možné jej převést do barevného modelu HSV (princip převodu je popsán v sekci 5.2). I když tato operace neupravuje samotné hodnoty optického toku, je možné, že konvoluční neuronová síť bude schopna lepšího odhadu při použití barevného modelu HSV. Porovnání obou výstupů ukazuje, že odhad rychlosti je u obou vstupů téměř totožný.

Zajímavějším experimentem je použití optického toku spolu s originálním snímkem videa. Zde je již potřeba upravit vstupní dimenze navržené konvoluční neuronové sítě a to tak, že dojde ke zvětšení počtu kanálů. Tedy optický tok a originální snímek jsou vloženy za sebe. Jak je vidět na obrázku 6.1, při použití kombinace optického toku a originálního snímku videa dojde k výraznému vyhlazení křivky odhadnuté rychlosti. Při aplikování postprocessingu je však samotný odhad rychlosti citelně horší.



Obrázek 6.1: Graf odhadu rychlosti, který porovnává vstup konvoluční neuronové sítě a to optický tok a optický tok v kombinaci s originálním snímkem videa. Na levém obrázku bez použití postprocessingu, na pravém obrázku je již postprocessing použit.

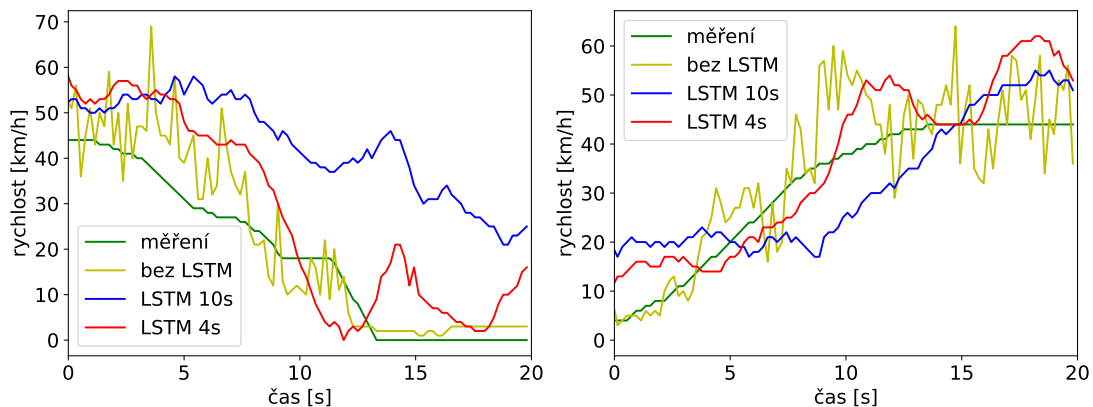
6.2 Rozšíření sítě o jednotku LSTM

Jak lze vidět na obrázku 5.5 výstup implementované konvoluční neuronové sítě je při pohledu na jednotlivé hodnoty značně zatížen chybou. Tato chyba je způsobena chybějícím kontextem v rámci delší sekvence snímků. Tedy konvoluční neuronová síť odhaduje rychlost pro jednotlivé snímky bez znalosti předešlých odhadů rychlosti a proto je možné vidět nerealistické skoky v odhadnutých rychlostech. Tento problém částečně řeší implementovaný postprocessing (obrázek 5.5), který průměruje odhadnutou rychlost v rámci delších sekvencí. Avšak toto řešení není ideální, neboť zejména v okrajových hodnotách dochází ke značnému zkreslení odhadu.

Lepším řešením může být rozšíření konvoluční neuronové sítě o jednotku LSTM (viz. sekce 3.3.1), která obsahuje i informaci o předešlých odhadech rychlosti a udržuje tedy žádaný časový kontext. Jednotka LSTM je vložena na konec stávající konvoluční neuronové sítě a pracuje tedy přímo s odhadnutými vektory rychlosti.

Jelikož jednotka LSTM pracuje nad časovými úseky sekvence, bude potřeba určit časovou délku jednoho úseku. Při volbě příliš krátké délky úseku bude mít výsledný kontext téměř nulovou vypovídající hodnotu a při volbě příliš dlouhého úseku může docházet ke zkreslování samotného odhadu rychlosti vozidla. Proto jsem se zaměřil na časové úseky o délce čtyř a deseti sekund.

Odhad rychlosti s využitím jednotky LSTM je zobrazen na obrázku 6.2. Jak lze vidět odhadnutý průběh rychlosti je daleko vyhlazenější a nedochází tedy k tak výrazným skokům mezi jednotlivými hodnotami rychlosti. Jednotka LSTM tedy může nahradit implementovaný postprocessing či značně redukovat velikost okna, s kterým postprocessing pracuje, což může vést k lepším výsledkům odhadu rychlosti. Zajímavé je také porovnání jednotlivých modelů s jednotkou LSTM a různými délkami časových úseků. Provedený experiment ukazuje, že úseky o délce deseti sekund jsou příliš „neohebné“ a tedy reagují hůře v situacích, kdy dochází ke změně rychlosti. Jako optimální délka časového úseku se jeví čtyři sekundy, které dosahují nejlepších výsledků.



Obrázek 6.2: Porovnání odhadu rychlosti pomocí modelů bez LSTM jednotky a s LSTM jednotkou o délce čtyř a deseti sekund.

6.3 Filtrace šumu v optickém toku

Odhadnutý optický tok vychází z běžných záběrů palubní kamery a tedy obsahuje mnoho nežádoucích informací. Příkladem mohou být pohybující se lidé, ostatní automobilová doprava či samotný pohyb vozidla, který nesouvisí s aktuální rychlostí vozidla. Pro odstranění těchto problémových informací jsem experimentoval s filtrováním či úpravou odhadnutého optického toku a poté porovnal výsledek s původním řešením.

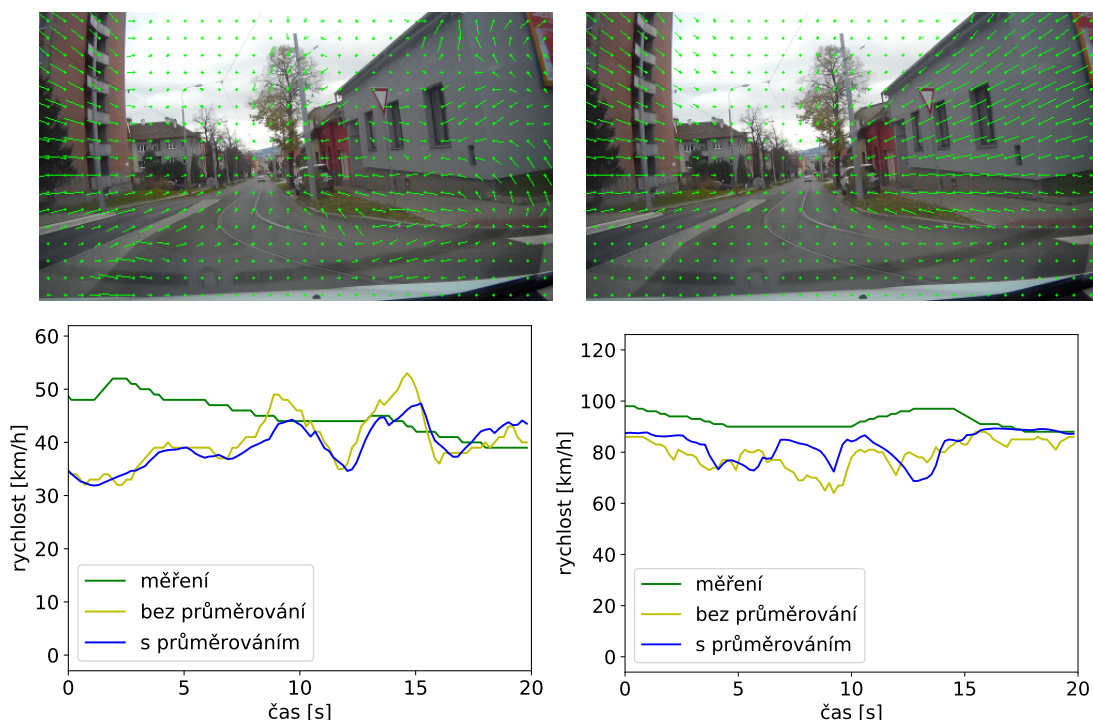
6.3.1 Průměrování optického toku

Nežádoucí pohyb kamery způsobený například přjetím hrbolu či díry v silnici ovlivňuje odhadnutý optický tok v rámci celého snímku. Tento pohyb se však v sekvenci snímků vyskytuje pouze lokálně. K vyfiltrování tohoto pohybu by tedy mohlo dojít v případě zprůměrování dostatečně velké sekvence matic optického toku. Navržený experiment však řeší pouze situaci, kdy je tento pohyb pouze lokální, tedy kupříkladu přejezd jednoho hrbolu. Naopak při častějším výskytu tohoto jevu dojde k zneprůhlednění části odhadů optického toku, které byly správné.

Výsledek průměrování optického toku je vidět na obrázku 6.3. Lze vidět, že zprůměrováním dojde k odstranění náhodných krátkých výkyvů v optickém toku, což je patrné zejména v pravé části obrázku. Porovnání odhadu rychlosti s použitím a bez použití průměrování optického toku je uveden na stejném obrázku. Jak je patrné z grafického znázornění, podařilo se mi docílit značného potlačení nežádoucího pohybu kamery, avšak samotný odhad rychlosti vozidla zůstává téměř beze změny. Dochází pouze k nepatrnému vyhlazení hodnot rychlosti.

6.3.2 Nulování optického toku

Odhad optického toku často obsahuje i ostatní automobilovou dopravu, která značně ovlivňuje odhad rychlosti vozidla. Tento jev je značně patrný například při zastavení vozidla na semaforech, zastínění výhledu palubní kamery velkým vozidlem či průchod osoby před vozidlem. Následující experimenty jsou tedy zacíleny na odfiltrování těchto nežádoucích pohybů z odhadnutého optického toku.



Obrázek 6.3: Grafické znázornění optického toku před a po průměrování v prvním řádku. Porovnání odhadu rychlosti bez použití průměrování optického toku a s jeho použitím na druhém řádku.

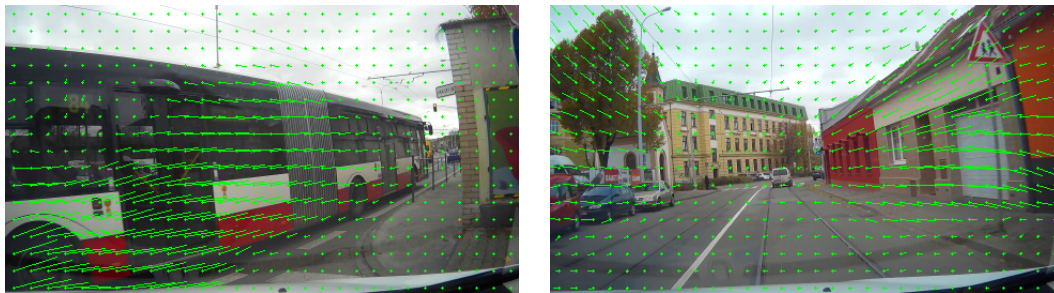
Nulování dle směru pohybu

Při zastavení vozidla na semaforech se v záběru palubní kamery ocitá střed křižovatky. Křižovatkou projíždí množství ostatních vozidel a tedy optický tok obsahuje jejich stopu, čímž vzniká pohyb i při nulové rychlosti vozidla. V této situaci se dá předpokládat, že směr pohybu ostatních vozidel bude horizontální či téměř horizontální, zatímco žádaný pohyb sledovaného vozidla je většinou vertikální. Jelikož matice optického toku obsahuje složku, která udává směr pohybu, je možné na základě tohoto údaje vybrat ty části, kde je pohyb horizontálního směru. Poté je na základě vybraných částí možné „vynulovat“ složku optického toku, která obsahuje informaci o síle pohybu.

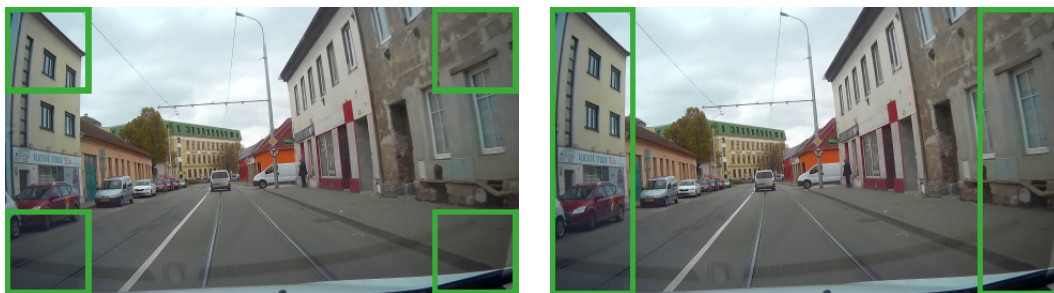
Provedené experimenty však ukázaly, že rozpětí vertikálního pohybu vozidel projíždějících křižovatkou značně proniká do rozpětí horizontálního pohybu zkoumaného vozidla. Dokonce i horizontální pohyb vozidla v některých částech obrazu pozvolně přechází do směru vertikálního, jak ilustruje obrázek 6.4. Tento způsob „nulování“ silové složky optického toku je tedy nevhodný.

Nulování dle síly pohybu

V případě výše nastíněné situace s křižovatkou nebo při rozjezdu vozidla stojícím před sledovaným vozidlem dochází ke značnému zkreslení žádané informace o rychlosti vozidla v odhadu optického toku. Tyto situace se však odehrávají především ve střední části snímku.



Obrázek 6.4: Znázornění optického toku, kdy je směr pohybu obdobný při kolmém průjezdu jiného vozidla před kamerou a ve standardní situaci.

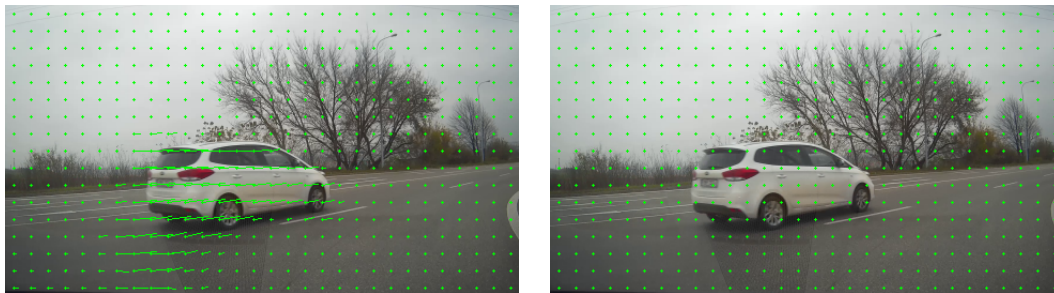


Obrázek 6.5: Vybrané oblasti pro určení žádaného pohybu v rámci celého snímku.

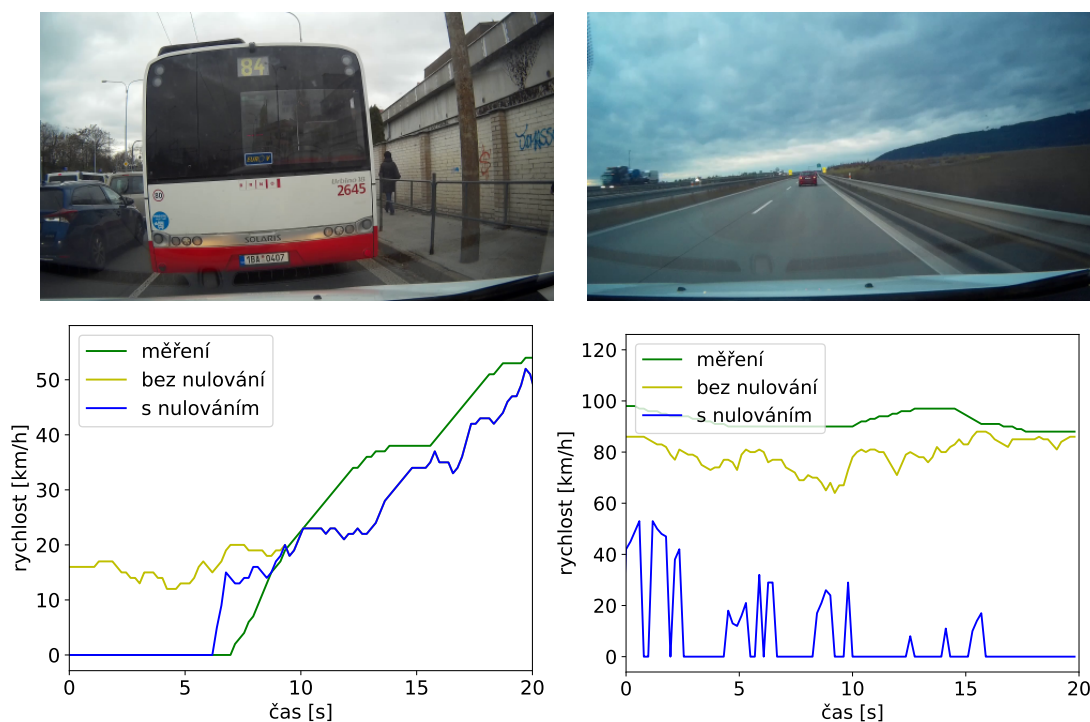
Optický tok, který je žádaný pro odhad rychlosti, nejčastěji vychází z nehybných objektů. V drtivé většině případů jsou těmito objekty domy či jiná zástavba v případě městského prostředí, či okolní krajina v případě opačném. Tyto objekty se zpravidla nacházejí v levé a pravé straně snímku. Lze tedy říci, že oblast středu snímku je méně významná. Je tedy možné vybrat okrajovou oblast snímku a dále ji zkoumat.

V rámci tohoto experimentu jsem vybral oblast v každém ze čtyř rohů snímku, jak je znázorněno na obrázku 6.5. Z vybraných oblastí je spočítán průměr hodnot silové složky optického toku a následně porovnán s určitou hodnotou prahu. Pokud je průměrná síla pohybu v daných oblastech menší než daná prahová hodnota, dochází k „vynulování“ celé silové složky optického toku. Tedy ve vybraných oblastech nedochází k pohybu či pouze k pohybu nepatrnému.

Jak lze vidět na obrázku 6.6 tento přístup je schopen eliminovat nežádoucí pohyb nacházející se ve středu snímku. Tedy tento přístup dokáže řešit situaci, kdy sledované vozidlo stojí na semaforech a v jeho výhledu projíždí jiná vozidla. Po důkladnějším zkoumání se však ukazuje, že při vyšších rychlostech (zejména na dálnici) je síla optického toku ve vybraných oblastech příliš slabá a její průměr padá pod určenou prahovou hodnotu. Po krátkém experimentování s hodnotou prahu je jasné, že při tomto přístupu nelze najít hodnotu, která je dostatečně malá pro úseky dálnice a zároveň dostatečně vysoká pro situaci křižovatky. Řešením může být rozšíření vybraných oblastí na celou výšku snímku, jak je znázorněno na obrázku 6.5.



Obrázek 6.6: Grafické znázornění optického toku před (vlevo) a po (vpravo) aplikování metody „nulování“ pomocí vybraných rohových (postranních) oblastí.



Obrázek 6.7: Náhled na první snímek videa a následný odhad rychlosti dané sekvence při použití metody „nulování“ pomocí vybraných rohových (postranních) oblastí při nižší a vyšší rychlosti.

I tento přístup je schopen odfiltrout kolmo projíždějící vozidla či rozjíždějící se vozidlo před sledovaným vozidlem. Ovšem i zde dochází při vyšších rychlostech ke stejnému problému i přesto, že zvolená hodnota prahu je „benevolentnější“.

V obou metodách jsem také pracoval se šířkou i výškou vybraných oblastí v rozmezí 5% až 20% šířky respektive výšky snímku, což také nevedlo k nalezení optimální hodnoty prahu.

Pro úplnost uvádím na obrázku 6.7 náhled na první snímek videa a naměřenou a odhadnutou rychlost dané sekvence. Sekvence umístěná na obrázku vlevo zachycuje situaci, kdy sledované vozidlo stojí za autobusem hromadné dopravy, který zakrývá téměř celý výhled.

Následně dochází k rozjezdu autobusu, což vede k zavedení velkých hodnot síly do odhadu optického toku a to způsobí velkou chybu v odhadu rychlosti, jak je vidět na přiloženém grafu. Druhá sekvence (vpravo) zachycuje běžnou jízdu na dálnici. Jak je vidět na přiloženém grafu odhadnuté rychlosti, tato metoda není použitelná při vyšších rychlostech vozidla.

6.4 Vyhodnocení systému

V této sekci jsou popsány, rozebrány a vyhodnoceny výsledky implementovaného systému pro odhad rychlosti. Nejprve je obecně vysvětlena struktura a význam uvedených výsledků spolu s popisem vyhodnocovaných chyb. Dále jsou rozebrány tři kategorie sekvencí – konstantní rychlost, akcelerace a decelerace. Zde je popsáno samotné video, diskutovány výsledky odhadu a porovnání při různých rychlostech. V další části je uvedena a vysvětlena průměrná procentuální chyba nad testovací datovou sadou. Ke konci sekce jsou uvedeny problémové situace a diskutována další práce.

6.4.1 Popis prezentovaných výsledků a metrik

Výsledky odhadu rychlosti vozidla jsou prezentovány formou kompozice obrázků a grafů. Uváděné obrázky (např. obrázek 6.9) jsou členěny do dvou sloupců, kde každý odpovídá jedné video sekvenci. Levý sloupec zachycuje nižší rychlosti zpravidla v městském prostředí, zatímco pravý sloupec představuje vyšší rychlosti zpravidla zachycené na dálnici či silnici I. a II. třídy.

Každý sloupec je dále členěn horizontálně. Na prvním řádku se nachází obrázek, který poskytuje náhled na prostředí dané video sekvence a měl by poskytnout představu o situaci zachycené v rámci daného videa. První uvedený graf zachycuje porovnání naměřené rychlosti a odhadnuté rychlosti jedním z uvedených modelů. Legenda grafu pak uvádí barevné kódování a u grafů obsahujících chyby i průměrnou chybu. Na třetím řádku se nachází graf, který uvádí chybu v kilometrech za hodinu pro jednotlivé použité modely. Tato chyba je počítána dle vzorce $e_a = v_e - v_t$, kde e_a je chyba, v_t je naměřená rychlost a v_e je odhadnutá rychlost. Na posledním grafu je uvedena relativní chyba, která značí procentuální rozdíl naměřené a odhadnuté rychlosti. Tato chyba je počítána dle vzorce $e_r = (1 - v_e \div v_t) * 100$, kde e_r je chyba, v_t je naměřená rychlost a v_e je odhadnutá rychlost. Vzhledem k dosažitelným hodnotám při rychlostech nižších než 10 kilometrů za hodinu a speciálně rychlosti 0 kilometrů za hodinu, je tato chyba shora omezena na 110%.

Odhad rychlosti vozidla je prováděn nad předem připravenými testovacími video sekvencemi o délce 30 sekund. Samotné vyhodnocení je provedeno nad posledními 20 sekundami, zatímco prvních 10 sekund slouží pro načtení dostatečně dlouhé sekvence pro průměrování optického toku a jednotku LSTM. Jednotlivé sekvence jsou vybrány tak, aby zachycovaly co nejrozmanitější rychlosti, prostředí a průběhy rychlostí.

6.4.2 Konstantní rychlost

První sloupec obrázku obsahuje video sekvenci, ve které je zachycen průjezd běžnou ulicí města. Blízko po obou stranách vozidla se nachází hustá zástavba po celou dobu videa. Okolní provoz je průměrný. Použité modely dokáží odhadnout rychlost poměrně přesně s minimální chybou. Modely také správně odhadují konstantní rychlost bez větších výkyvů.

Z vyhodnocení relativní chyby vychází model s jednotkou LSTM o něco lépe s průměrnou chybou 5%.

Video sekvence ve druhém sloupci zachycuje standardní jízdu po dálnici. V blízkosti dálnice jsou umístěna svodidla, ve střední vzdálenosti občas několik keřů a v dáli rozlehlá krajina. Okolní provoz je průměrný. Použité modely v první části sekvence odhadují rychlost s poměrně obdobnou úspěšností. Ve druhé polovině však model s jednotkou LSTM detekuje značný výkyv v rychlosti. Toto je pravděpodobně způsobeno průjezdem mezi dvěma jinými automobily po obou stranách vozidla, což lze také vidět na přiloženém obrázku. Vyhodnocované modely podhodnocují skutečnou rychlost o zhruba 15 kilometrů za hodinu a dosahují relativní chyby kolem 13%.

Výsledky pro konstantní rychlosti jsou uvedeny na obrázku 6.9.

6.4.3 Klesající rychlost

Sekvence uvedená v prvním sloupci zachycuje příjezd na parkoviště a následné parkování na kolmém stání. Levá část snímku obsahuje blízkou zástavbu, v pravé části jsou zachycena parkující vozidla. V posledních sekundách video sekvence probíhá parkování na kolmém stání, kdy se v záběru vyskytuje blízko zaparkované vozidlo. Okolní provoz je nulový a panuje mlhavé počasí. Použité modely odhadují rychlost téměř totožně, výjimkou je konec sekvence, kde dochází u modelu s jednotkou LSTM k odhadu vyšší rychlosti. Toto chování je pravděpodobně způsobeno pomalým ostrým zatočením v kombinaci s velmi blízko umístěným objektem. V grafu absolutní chyby lze vidět, že odhadnutá rychlost je nadhodnocena až o 10 kilometrů za hodinu. Relativní chyba pak dosahuje vysokých hodnot kvůli velmi nízkým rychlostem.

Druhý sloupec obsahuje video sekvenci pořízenou v pozdějších hodinách, kdy se již stmívá. Zachycena je situace sjezdu z dálnice, po obou stranách jsou umístěna svodidla a za nimi nepřilís vzrostlé stromy. V dáli je výhled na krajinu. Použité modely se v začátku sekvence shodují v odhadu rychlosti, avšak model s jednotkou LSTM poté reaguje daleko později na snížení rychlosti. Ke konci sekvence se již chovají obdobně. Model bez jednotky LSTM je schopen správně detekovat klesající rychlost. Samotný odhad rychlosti je nadhodnocen až o 50 kilometrů za hodinu, průměrná relativní chyba pak také dosahuje vysokých hodnot.

Výsledky pro klesající rychlosti jsou uvedeny na obrázku 6.10.

6.4.4 Stoupající rychlost

Video sekvence uvedená v prvním sloupci zachycuje jízdu po městském okruhu. Na začátku sekvence je v téměř celém výhledu kamery zachycen rozjíždějící se autobus (viz. obrázek 6.7 nahoře vlevo). Ve střední vzdálenosti od vozidla se nachází výhled na okolní krajinu. Okolní provoz je hustý. Použité modely odhadují rychlost obdobným způsobem, model bez jednotky LSTM však rychleji reaguje na změnu rychlosti. Značnou chybu, která je patrná v první části sekvence, způsobuje výše popsaný rozjezd autobusu, který pokrývá téměř celý snímek. Relativní chyba dosahuje hodnot okolo 40%, což je způsobeno nízkými rychlostmi v této video sekvenci.

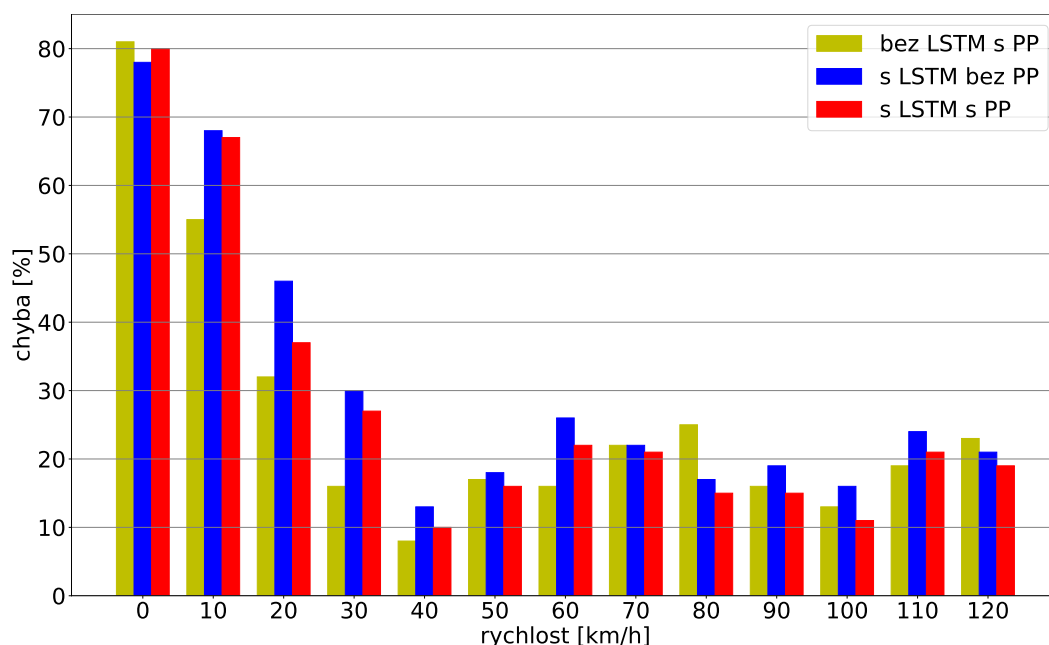
Druhá video sekvence zachycuje příjezd na dálnici v lehce mlhavém počasí. Po obou stranách vozidla jsou v blízké vzdálenosti svodidla, ve střední vzdálenosti se vyskytuje vegetace a v dáli je výhled do krajiny. Okolní doprava je průměrná až lehce podprůměrná. Hlavní rozdíl v odhadu rychlosti jednotlivých modelů je v samotném začátku sekvence, kde se jejich odhad liší o zhruba 30 kilometrů v hodině. Reakce na změnu rychlosti je obdobná,

i když model bez jednotky LSTM se zdá být reaktivnější. Průměrná chyba modelů se pohybuje kolem 15 - 20 kilometrů za hodinu a relativní chyba kolem 17%.

Výsledky pro zvyšující se rychlosti jsou uvedeny na obrázku 6.11.

6.4.5 Průměrná chyba

Závěrečným vyhodnocení této práce je průměrná relativní chyba v rámci testovací datové sady. Tato chyba je vyhodnocena po úsecích o velikosti deset kilometrů za hodinu. Výsledný graf je uveden na obrázku 6.8. Jak lze vidět, chyba při nižších rychlostech dosahuje vysokých hodnot. Speciálním případem je pak úsek rychlostí 0 až 5 kilometrů za hodinu, kde chyba dosahuje až 80%. Naopak zhruba od rychlosti 35 kilometrů za hodinu se dá říci, že implementovaný systém je schopný odhadnout rychlost s 20% chybou. Z grafu lze také vidět, že model bez jednotky LSTM a s použitím postprocessingu dosahuje v rámci tohoto vyhodnocení při nižších rychlostech lepších výsledků. Toto je pravděpodobně způsobeno svižnější reakcí na změnu rychlosti, která je v intervalu 0 až 35 kilometrů za hodinu zpravidla velice rychlá. Při vyšších rychlostech je již chybovost obou modelů obdobná.



Obrázek 6.8: Graf zachycuje průměrnou relativní chybu u vyhodnocovaných modelů. Chyba je počítána nad úseky o délce 10 km/h, tedy například chyba pro rychlost 20 km/h je počítána z úseku 15 až 25 km/h.

6.4.6 Problematické situace

Jedním z nejproblematictějších faktů jsou bezesporu další pohyblivé objekty v okolí vozidla. Těmito objekty jsou ve většině případů ostatní vozidla, která zastiňují výhled kamery na statickou scénu. A také zavádí jejich vlastní pohyb do odhadu optického toku. Z výsledků práce vyplývá, že navržené řešení se dokáže relativně uspokojivě vyrovnat s okolním provozem, pokud je samotné sledované vozidlo v pohybu. Horší situace nastává, pokud sledované

vozidlo stojí a před objektivem kamery se jiné vozidlo rozjíždí či projíždí. V tomto případě je již odhad značně nepřesný.

Další problematická situace nastává v případě těsné blízkosti objektu a vozidla. Tato situace může nastat například při průchodu chodce před vozidlem, či při kolmém parkování vozidla. V této situaci je odhadnutý optický tok velice silný a zaujímající většinu sledovaného prostoru, čímž dojde k nepřesnému odhadu rychlosti.

V případě porovnání nižších a vyšších rychlostí je absolutní chyba poměrně rozlišná. Avšak procentuální chyba je relativně stejná, což značí, že navržený systém je schopen odhadnout rychlost vozidla konzistentním způsobem v obou případech.

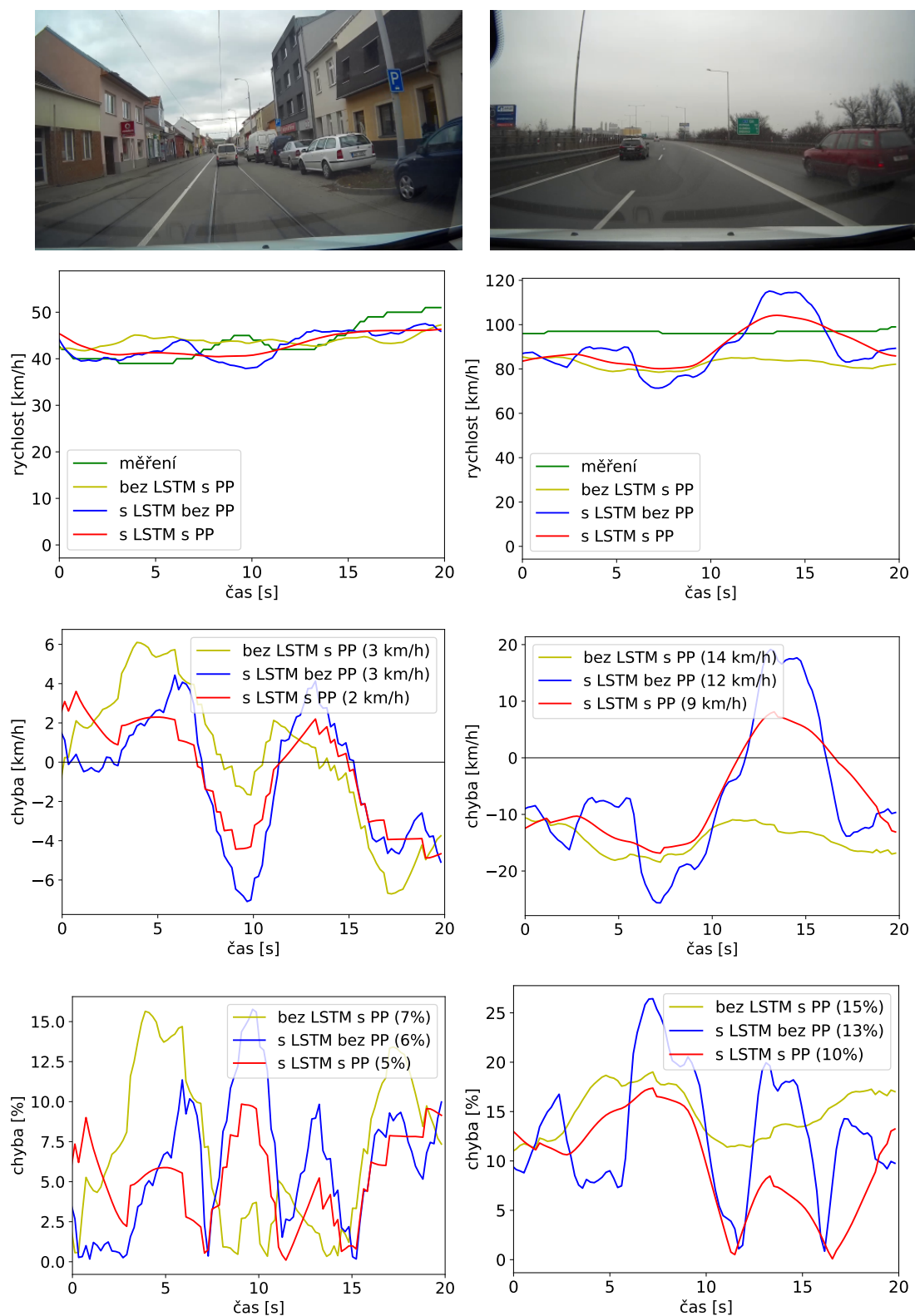
6.4.7 Další práce

Další pokračování této práce se může věnovat řešení výše zmíněných problémů. Pro odfiltrování okolního provozu, které značně zkresluje optický tok, může být použita například sémantická segmentace. Sémantická segmentace spočívá v rozdělení obrázku do více částí, které odpovídají určité třídě objektů. Tímto by bylo možno vybrat z obrazu pouze statické objekty jako jsou budovy či okolní krajina a zároveň odstranit pohyblivé objekty.

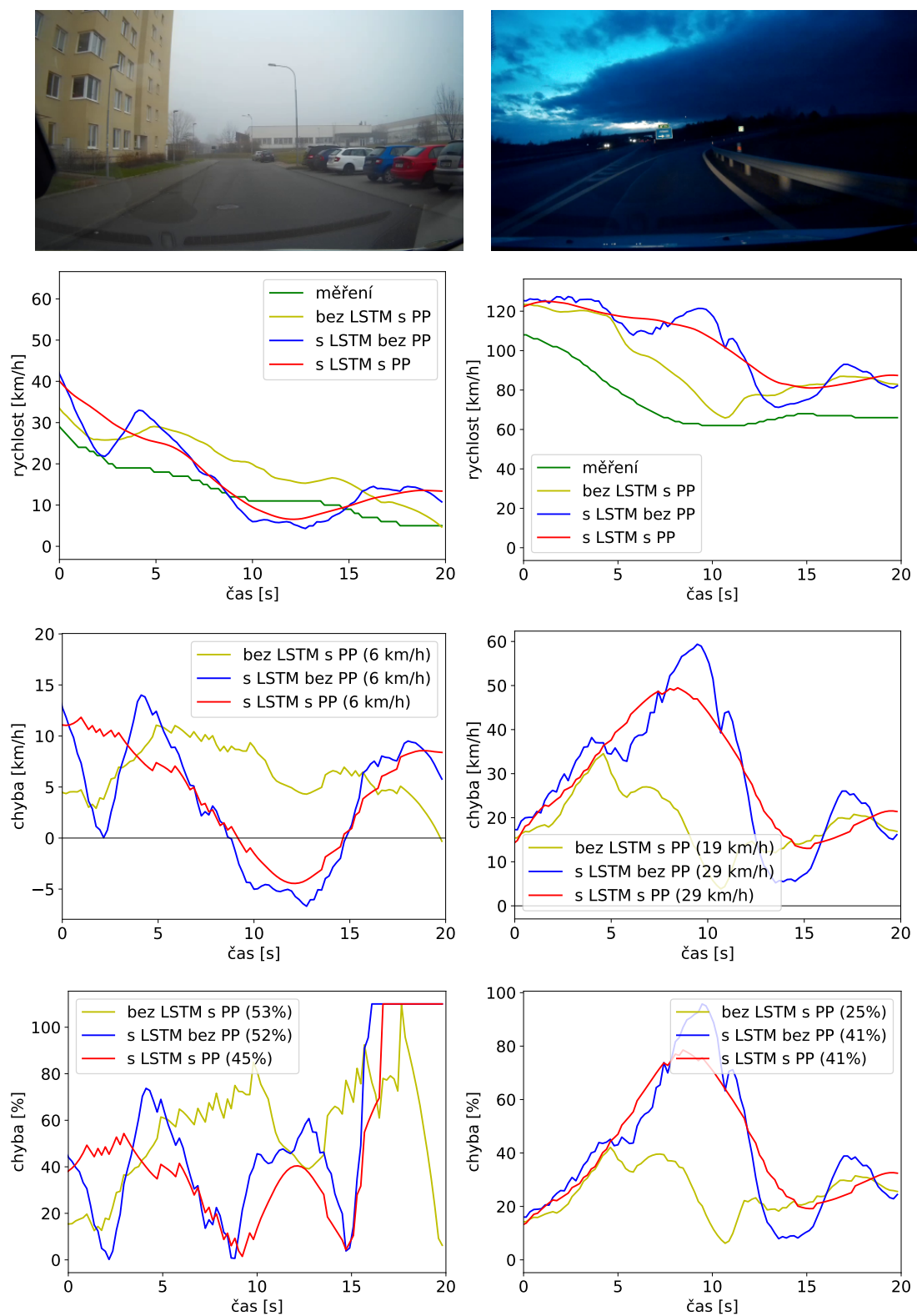
Dále by bylo možné rozšířit vstupní data konvoluční neuronové sítě o odhad hloubky v obraze. Pro tento účel je možno využít například přístup z článku [9]. Odhad hloubky by mohl přinést nové informace o snímku a tím vylepšit výsledný odhad rychlosti.

Další rozšíření by mohlo spočívat v použití konvoluční neuronové sítě FlowNet [13], která slouží pro odhad optického toku. Tento odhad by měl být přesnější než použitý algoritmus Dense Inverse Search a tím pádem by měl být výsledný odhad rychlosti přesnější.

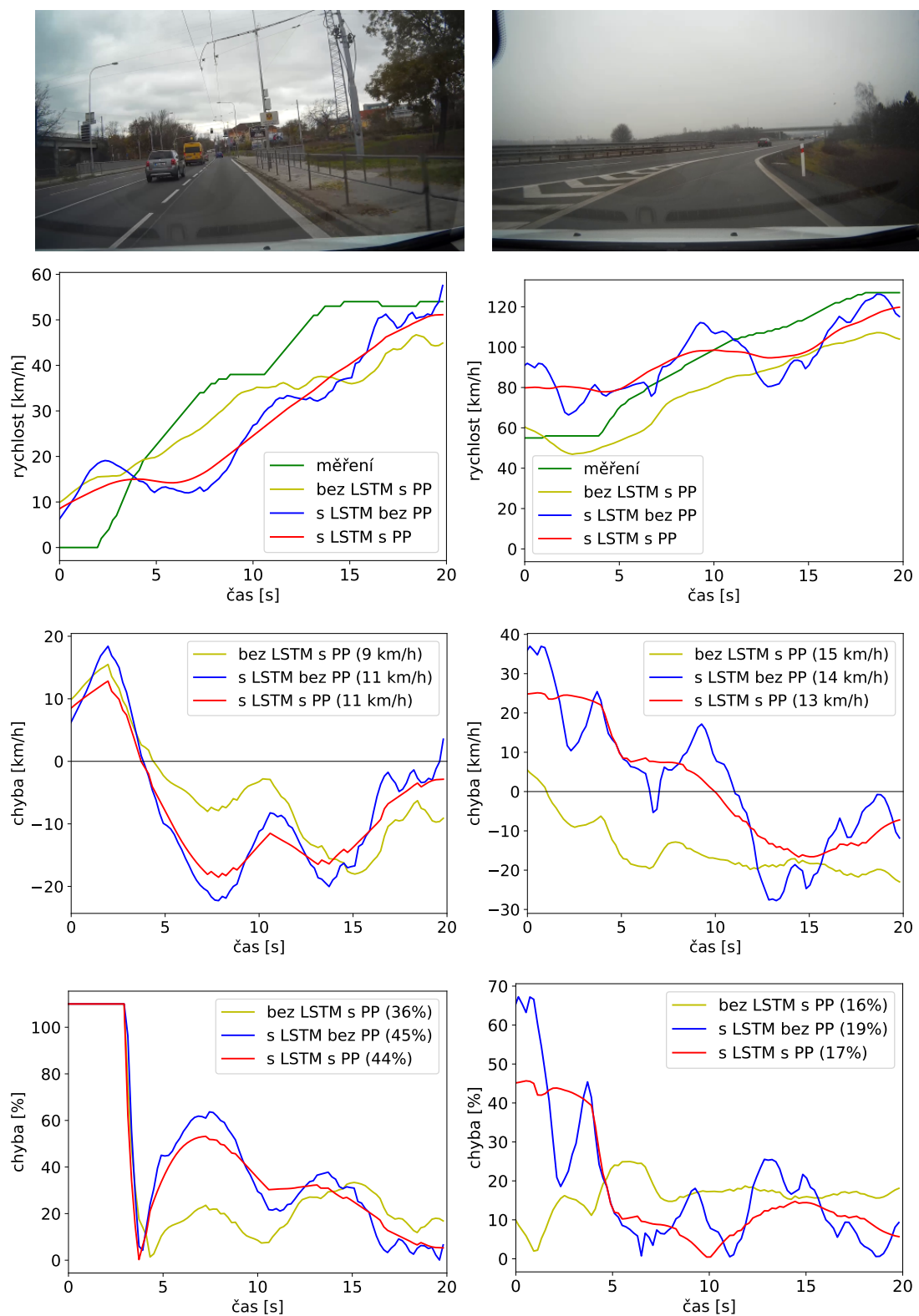
Samozřejmostí je pak další rozšiřování datové sady. Sada vytvořená v rámci této práce byla pořízena v podzimním období, jednou kamerou, jedním vozidlem a na stejném území. Rozšiřování by tedy mělo spočívat v zavedení další různorodosti v rámci těchto parametrů.



Obrázek 6.9: Vyhodnocení odhadu rychlosti nad sekvencemi obsahujícími konstantní rychlost. Detailnější popis struktury obrázku a grafů je uveden v sekci 6.4.1 a popis dosažených výsledků v sekci 6.4.2.



Obrázek 6.10: Vyhodnocení odhadu rychlosti nad sekvencemi obsahujícími snižující se rychlost. Detailnější popis struktury obrázku a grafů je uveden v sekci 6.4.1 a popis dosažených výsledků v sekci 6.4.3.



Obrázek 6.11: Vyhodnocení odhadu rychlosti nad sekvencemi obsahujícími zvyšující se rychlost. Detailnější popis struktury obrázku a grafů je uveden v sekci 6.4.1 a popis dosažených výsledků v sekci 6.4.4.

Kapitola 7

Závěr

V rámci této diplomové práce jsem se seznámil s problematikou odhadu rychlosti vozidla z palubní kamery a prozkoumal různé možnosti řešení. Z existujících metod jsem vybral odhad rychlosti pomocí optického toku a konvoluční neuronové sítě a následně jsem se detailněji seznámil s těmito metodami.

Dále jsem provedl prvotní implementaci problému a její validaci, která ukazuje, že zvolený přístup je schopen odhadu rychlosti vozidla ze záznamu palubní kamery. Odhadnutá rychlost byla však zatížena velkým množstvím chyb vyplývajících ze složitosti zkoumané scény. Za účelem jejich odstranění jsem provedl několik experimentů, které zpřesnili odhad rychlosti vozidla.

Nakonec práce jsem se zabýval vyhodnocením implementovaného systému, z kterého vyplývá, že odhad rychlosti pomocí zvoleného a implementovaného přístupu je schopen odhadnout rychlost s průměrnou chybou 20% na úsecích s rychlostí vyšší než 35 kilometrů za hodinu. Úseky s nižší rychlostí jsou již zatíženy značnou průměrnou chybou, která je nejvyšší při rychlosti 0 kilometrů za hodinu, což také z části vyplývá z použitého výpočtu.

Dalším pokračováním práce může být prozkoumání sémantické segmentace, pomocí které by bylo možno odfiltrovat okolní provoz nebo použití dalšího vstupu, kterým může být například odhad hloubky ve snímku. Vylepšení samotného odhadu rychlosti by mohlo přinést použití konvoluční neuronové sítě pro odhad optického toku.

Literatura

- [1] Aqel, M. O.; Marhaban, M. H.; Saripan, M. I.; aj.: Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, ročník 5, č. 1, 2016: str. 1897.
- [2] Barbosa, R.; Silva, J.; Junior, M. M.; aj.: Velocity estimation of a mobile mapping vehicle using filtered monocular optical flow. In *International Symposium on Mobile Mapping Technology*, ročník 5, 2007.
- [3] Bevilacqua, M.; Tsourdos, A.; Starr, A.: Egomotion estimation for monocular camera visual odometer. In *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*, IEEE, 2016, s. 1–6.
- [4] Bojarski, M.; Del Testa, D.; Dworakowski, D.; aj.: End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.
- [5] Bradski, G.; Kaehler, A.: *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.
- [6] Chhaniyara, S.; Bunnun, P.; Seneviratne, L. D.; aj.: Optical flow algorithm for velocity estimation of ground vehicles: A feasibility study. *International Journal on smart sensing and intelligent systems*, ročník 1, č. 1, 2008: s. 246–268.
- [7] Daubner, L.: *Deep learning*. Bakalářská práce, Masarykova univerzita, 2015.
URL https://is.muni.cz/th/410034/fi_b/thesis.pdf
- [8] Deeplearning4j: A Beginner's Guide to Recurrent Networks and LSTMs. [Online; navštíveno 19.3.2018].
URL <https://deeplearning4j.org/lstm.html>
- [9] Godard, C.; Mac Aodha, O.; Brostow, G. J.: Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, ročník 2, 2017, str. 7.
- [10] Han, I.: Car speed estimation based on cross-ratio using video data of car-mounted camera (black box). *Forensic science international*, ročník 269, 2016: s. 89–96.
- [11] Hochreiter, S.; Schmidhuber, J.: Long short-term memory. *Neural computation*, ročník 9, č. 8, 1997: s. 1735–1780.
- [12] Hradiš, M.: Konvoluční neuronové sítě. OpenAlt konference, 2015, [Online; navštíveno 12.3.2018].
URL <https://www.superlectures.com/openalt2015/konvolucni-neuronove-site>

- [13] Ilg, E.; Mayer, N.; Saikia, T.; aj.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ročník 2, 2017.
- [14] Kampelmuehler, M.; Mueller, M.; Feichtenhofer, C.: Camera-based Vehicle Velocity Estimation using Spatiotemporal Depth and Motion Features. 2017, [Online; navštíveno 6.1.2018].
URL http://benchmark.tusimple.ai/static/files/poster_velocity_1.pdf
- [15] Karpathy, A.: Convolutional Neural Networks for Visual Recognition. 2017, [Online; navštíveno 27.12.2017].
URL <http://cs231n.github.io/convolutional-networks/>
- [16] Kroeger, T.; Timofte, R.; Dai, D.; aj.: Fast optical flow using dense inverse search. In *European Conference on Computer Vision*, Springer, 2016, s. 471–488.
- [17] Mitchell, J.: Autonomous Vehicle Speed Estimation from dashboard cam. 2017, [Online; navštíveno 5.1.2018].
URL <https://chatbotslife.com/autonomous-vehicle-speed-estimation-from-dashboard-cam-ca96c24120e4>
- [18] Olah, C.: Understanding LSTM Networks. 2015, [Online; navštíveno 19.3.2018].
URL <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [19] Šonka, M.; Hlaváč, V.; Boyle, R.: *Image processing, analysis, and machine vision*. Cengage Learning, 2014.
- [20] Reichl, J.; Všetická, M.: Rychlost hmotného bodu. [Online; navštíveno 21.12.2017].
URL <http://fyzika.jreichl.com/main.article/view/6-rychlost-hmotneho-bodu>
- [21] Veličković, P.: Deep learning for complete beginners: convolutional neural networks with keras. 2017, [Online; navštíveno 10.2.2018].
URL <https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html>
- [22] Wang, S.; Clark, R.; Wen, H.; aj.: Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, 2017, s. 2043–2050.
- [23] Wedel, A.; Cremers, D.: *Stereo scene flow for 3D motion analysis*. Springer Science & Business Media, 2011.
- [24] Wikipedie: Autokamera. 2016, [Online; navštíveno 1.5.2018].
URL <https://cs.wikipedia.org/wiki/Autokamera>
- [25] Yixuan, H.: Loss Functions In Deep Learning. 2017, [Online; navštíveno 12.3.2018].
URL <http://yeephycho.github.io/2017/09/16/Loss-Functions-In-Deep-Learning/>